

INTRODUCTION TO MACSYMA

Introduction to Macsyma

Technical Documentation Staff
Macsyma Inc.

This document corresponds to Macsyma Release 2.2 for Windows

The Technical Documentation Staff at Macsyma Inc. prepared this document. The software described in this document is proprietary to, and comprises valuable trade secrets of, Macsyma Inc. It is given in confidence by Macsyma pursuant to a written license agreement, and may be used, copied, transmitted, and stored only in accordance with the terms of such license. This document may not be reproduced in whole or in part without the prior written consent of Macsyma Inc.

Copyright © 1982-1997 Macsyma Inc. All Rights Reserved. Portions Copyright © 1982 Massachusetts Institute of Technology. All Rights Reserved.

Macsyma®, and PDEase® are registered trademarks of Macsyma Inc. DataViewer™, MathTips™, NumKit™ and PDEase2D™ are trademarks of Macsyma Inc. All other trademarks are the property of their respective owners.

All product names used herein which are not the trademarks of Macsyma Inc. are the trademarks of their respective owners.

Restricted Rights Legend

Use, duplication, and disclosure by the Government are subject to restrictions as set forth in subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at FAR 52.227-7013.

Macsyma Inc.
20 Academy Street
Arlington, MA 02174
U.S.A.

TEL: 617-646-4550
FAX: 617-646-3161

Electronic Mail:

service@macsyma.com

info-macsyma@macsyma.com

URL <http://www.macsyma.com>

Printed in the United States of America.

Print year and number: 97 1

Contents

1. START MACSYMA	1
2. ON-LINE HELP WITH THE MATH TIPS ADVISOR AND NATURAL LANGUAGE QUERY	5
3. ENTER COMMANDS	16
4. BUILD MACSYMA NOTEBOOKS	23
5. DO MATHEMATICS WITH MACSYMA	34
6. MANIPULATE MACSYMA GRAPHICS	52
7. MEET THE DATAVIEWER™ AND MFE_DATA PACKAGE	56
8. ANSWERS TO SOME COMMON QUESTIONS	67
9. FURTHER ASSISTANCE	74

Welcome to Macsyma. Macsyma is a large computer algebra system capable of:

- manipulating symbolic expressions in algebra and calculus
- drawing 2D and 3D plots and graphs to visualize relationships
- computing and viewing numbers arising in mathematical problems.

Using Macsyma, you can differentiate, integrate, take limits, solve equations, factor polynomials, expand functions in power series, take Laplace transforms and their inverses, manipulate tensors, and plot in two and three dimensions.

The Macsyma Student Edition gives you the full power of the Professional Version, but restricts the size of certain types of problems:

1. You may build a matrix up to 10 x 10.
2. You may enter Command of up to 100 lexemes.
3. You may use bfloat precision up to 32 digits.
4. You may plot up to 100 points in two dimension and up to 20 points in three dimensions.

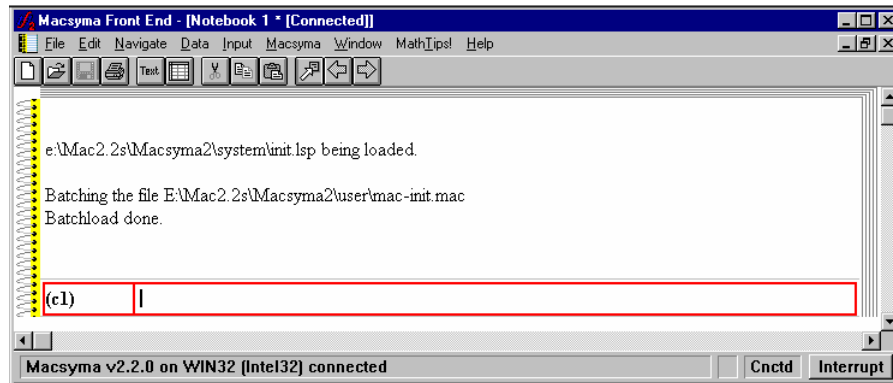
Advanced notebook capabilities, translation and compilation of Macsyma code, and access to the Lisp system in which Macsyma is written are not supported.

The entire online Help system with MathTips™ and MathHelp is included with the Student Edition.

The *Macsyma Reference Manual* and *Macsyma User's Guide* may be purchased separately.

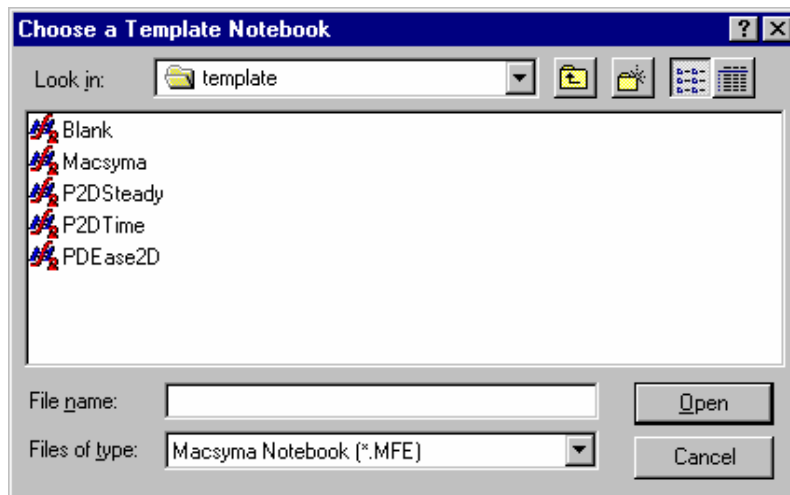
1. START MACSYMA

After you have installed Macsyma, start the program by double-clicking on the Macsyma icon in the Macsyma folder. You should automatically open the Macsyma Front End window, a Macsyma notebook document, and connect the notebook document to a Macsyma Math Engine. (The word “Connected” appears at the top of the notebook.) The MFE icon opens a Macsyma Front End window, in which you can load and edit a notebook, or open a notebook template without connecting the Macsyma Math Engine. Connecting the Front End to a Math Engine will produce a notebook that looks something like:¹

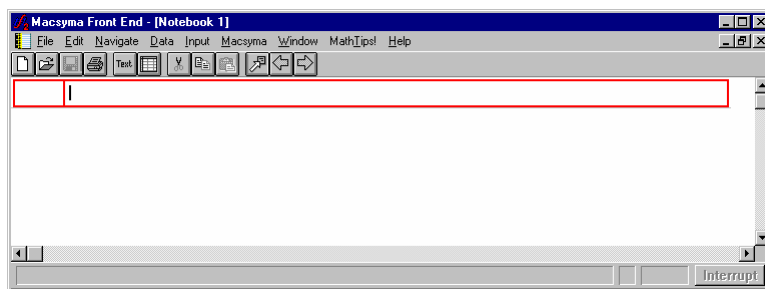


The Macsyma Front End program also serves as the Front End for PDEase2D. You can select PDEase2D, Macsyma, or blank template notebooks.

¹ All illustrations of Macsyma 2.2 screens were made using the Windows 95 Explorer interface, but the basic components of each screen are the same across all three Windows OS.



Select the Macsyma template notebook to open an empty Macsyma notebook. You will see a notebook that looks like this:

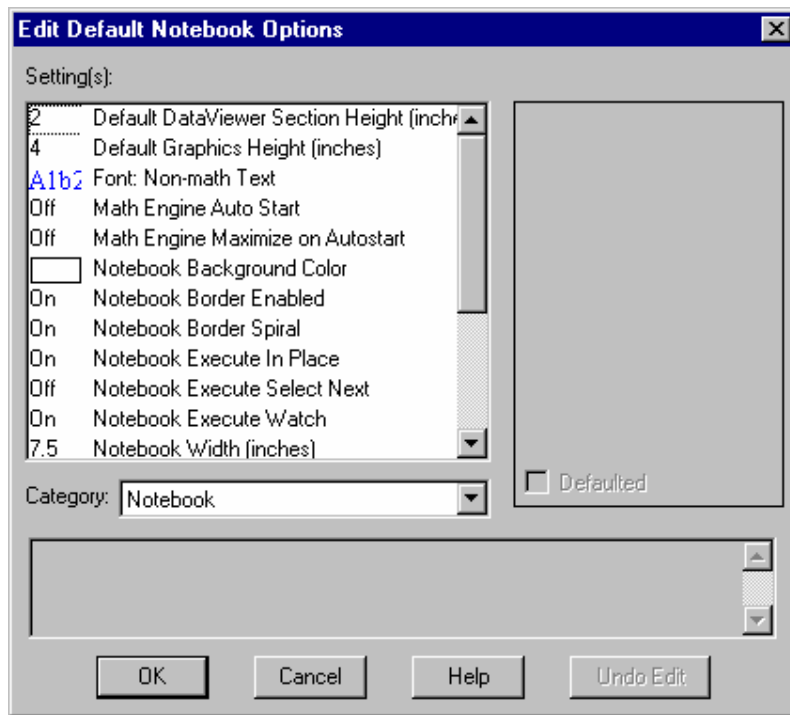


You can connect this empty notebook to the Macsyma Math Engine by:

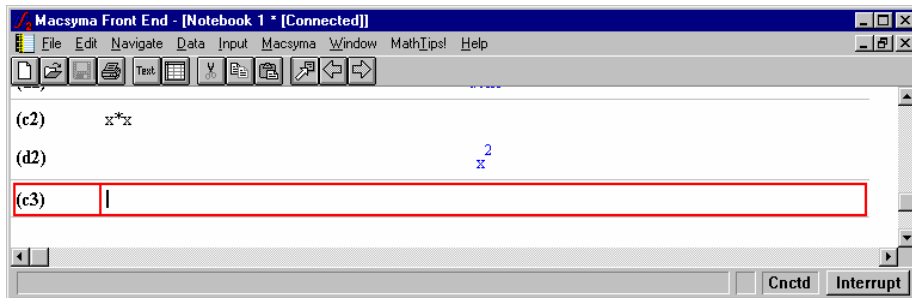
1. Typing an expression to evaluate (*e.g.*, $x*x$ followed by Enter). Macsyma opens a dialog box to open a math server. Select New Server.
2. Clicking on Macsyma; you can connect from the Menu Bar.

The Macsyma notebook is structured as follows:

1. Window Title (Macsyma Front End)
2. Menu Bar (File Edit Navigate ...)
3. Button Bar, which offers context appropriate actions and adjusts automatically to each context



Without the spiral border, section brackets, and borders, the notebook looks like:

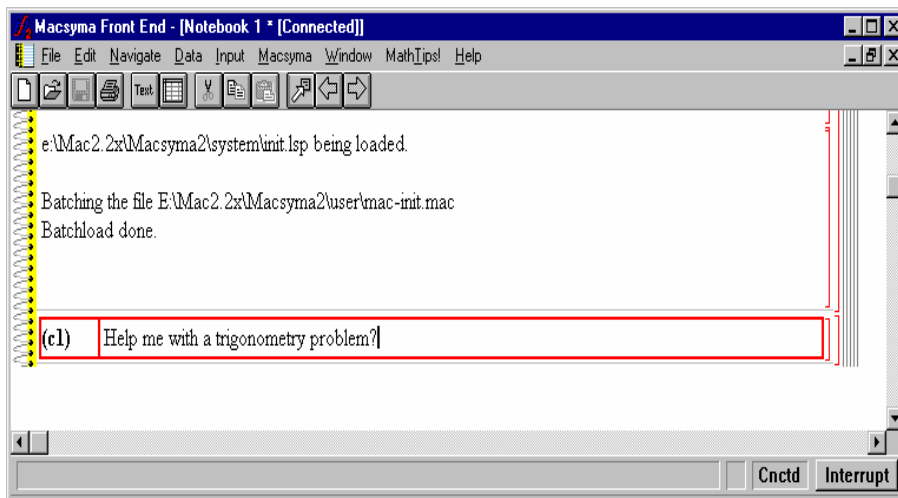


2. ON-LINE HELP WITH THE MATH TIPS ADVISOR AND NATURAL LANGUAGE QUERY

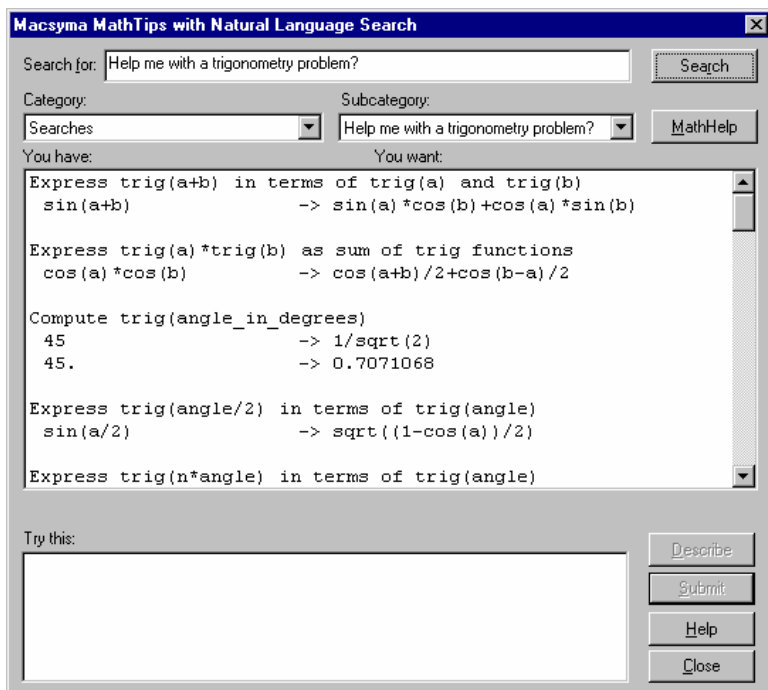
2.1. The MathTips™ Advisor

When you know the mathematical operation you want to perform, but don't know the correct command, you can use Help - MathTips natural language query to narrow your search, and inspect the different topics and sub-topics to perform computations.

Suppose you want to solve a trigonometric problem. You can type on the input line "Help me with a trigonometry problem?"



and then hit Enter. MathTips will respond with:



The main area of the advisor is filled with tips on common computations. Each tip consists of:

- A one-line verbal description of the computational task;
- An example input to the computation, which appears on the left;
- The desired output from the computational task, which appears on the right;
- When you select a tip with your mouse, the command or short program that accomplishes the computational task appears in the area at the bottom of the tips advisor.

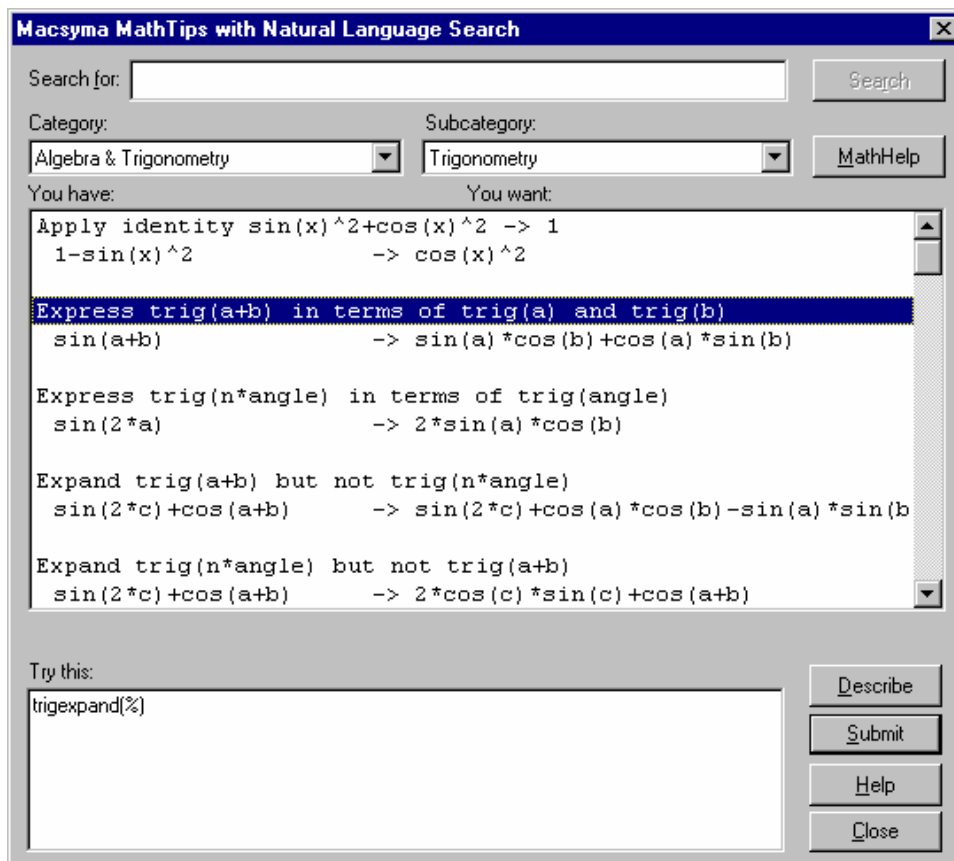
You can inspect each of the tips and find the commands that are relevant to the particular calculation you want to do. You can see several relevant tips. You can now return to your Macsyma screen and type the expression you want to solve, *e.g.* simplify cot(x+y).

You can now use the **MathTips** advisor to solve this trigonometric problem. To expand the trigonometric function $\cot(x+y)$, type the following expression on your current Macsyma command line and press the **Enter** key:

(c1) $\cot(x+y)$

(d1) $\cot(y+x)$

Now open the **MathTips** advisor by clicking on **Help - MathTips** in the **Help** menu. Click in the box in the upper left of the advisor, then on the topic **Algebra & Trigonometry**, and then on the subtopic **Trigonometry**. The **MathTips** advisor will look like this:



One of the tips says:

Express $\text{trig}(a+b)$ in terms of $\text{trig}(a)$ and $\text{trig}(b)$
 $\sin(a+b) \quad \rightarrow \quad \sin(a)*\cos(b)+\cos(a)*\sin(b).$

When you click on this tip, the command for accomplishing this transformation appears in the bottom area of the advisor:

```
trigexpand(%)
```

where % refers to the left-hand expression in the tip, $\sin(a+b)$ and to the previous output line in Macsyma.

You can apply this command to your previous output, $\cot(y+x)$, without re-typing it. By clicking on the button **Submit**, you submit the command `trigexpand(%)` to Macsyma. The new command line is selected in red, so pressing the **Enter** key executes this command, yielding the result:

(c2) trigexpand(%)

(d2)
$$\frac{\cot(x) \cot(y) - 1}{\cot(y) + \cot(x)}$$

Select a subtopic; the **MathTips** advisor presents you with a list of common computations in that area. The advisor also allows you to visit **MathHelp!** Topics browser. See Section “2.2. The **MathHelp!** Topic Browser.”

You can perform many computations using the **MathTips** advisor by copying and modifying the **MathTips** examples; so you rarely need to read command descriptions to determine which commands to use.

You can also open the **MathTips** advisor from **MathHelp!** browser inside the mathematics topic browser using the **Tips** button or by clicking **Help - MathTips** in the **Help** menu.

You can also click on the **Help** button for “**Tips on Tips.**”

If you try to follow the guidelines below while you compose your queries, you can improve the effectiveness of your interactions with the system:

- Avoid using the personal pronoun “I”, which the system sometimes interprets as $i = \sqrt{-1}$.
- Carefully select mathematically significant words, paying attention to the differences between polynomials and equations, and mathematical constants and physical constants.

- Use verbs that describe the operation, such as “solve,” “simplify,” “factor,” or “expand.”

- Avoid mathematically imprecise words; they may inadvertently distort the relative ranking of the tips for reasons that have no relation to the problem you want to specify. After MathTips supplies you with recommended tips, you can benefit from doing a little exploring:

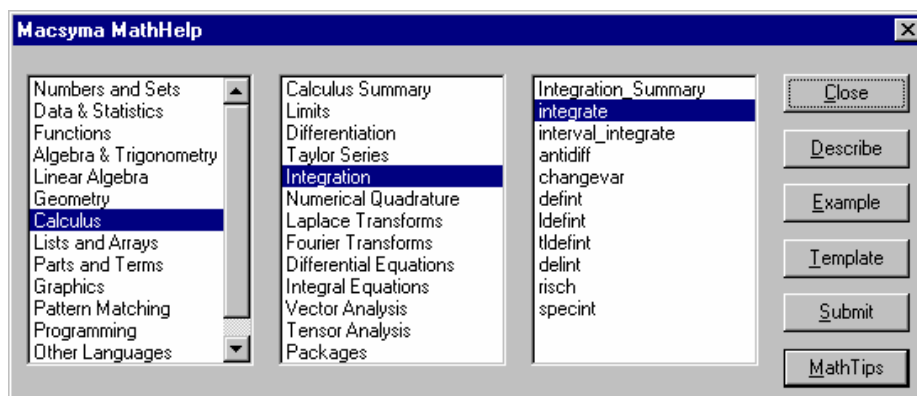
- Scroll through the tips. Macsyma tries to put the right tip in the first or second position, but you may find a better tip 5-10 places down on the list.

- Try variations on the words and the word endings. For example, you might ask to “simplify a product of complex expressions” or “write a product of complex expressions in rectangular form,” both of which work in the MathTips Advisor.

2.2. The MathHelp! Topic Browser

The menu bar at the top of the Macsyma Front End window resembles the preceding illustration. With your mouse, click left on **MathHelp!**²

You will see a topic browser dialog box containing the items in the left-hand column in the figure below. Click on **Calculus**. A list of calculus subtopics appears in the second column, as in the figure below. Click on **Integration**. The third column of the dialog box will be filled with a partial list of Macsyma's integration commands. The dialog box will look much like the one in the illustration:



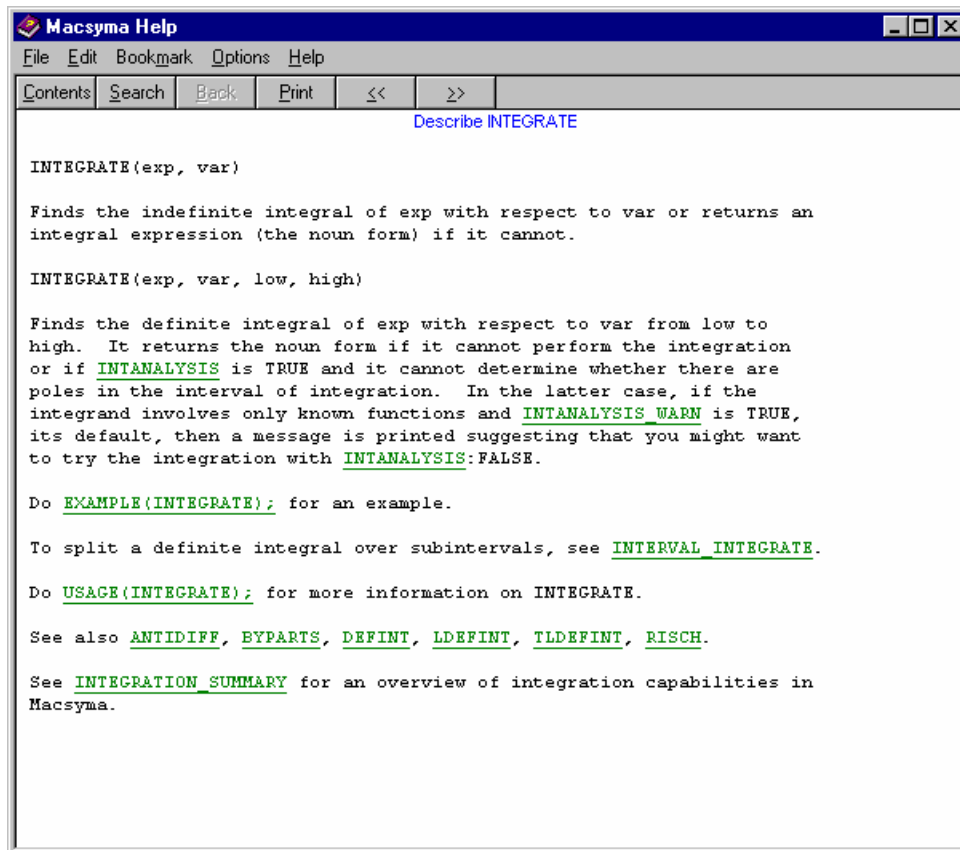
² Throughout this document, click with the left mouse button unless the instructions explicitly state otherwise.

Scroll through the third column to expose more of Macsyma's integration commands. The first item is a summary of integration commands. The next command on the list, Integrate, is the most commonly used command in this group. Click on this command name now.

When you know the mathematical operation you want to perform, but don't know the correct command, you can choose **Tips** to use the MathTips Advisor to help narrow your search by showing Macsyma commands and their behavior.

2.3. Hypertext Descriptions of Macsyma Commands

Click on the **Describe** button for a description of the command in the Macsyma Help window. The result provides the command's calling syntax, an explanation of what the command does, and cross references. This description is "hypertext;" some words (which are colored and/or underlined in the text) have hot links to other parts of the text. You will see a Help Window that looks like:



- Click on the phrase `USAGE(INTEGRATE)`; for a detailed discussion of the **integrate** command. After inspecting this screen, click on **Back** on the top menu bar of the Macsyma Help window to return to the previous screen, a description of **integrate**.
- At the end of the description of the **integrate** command, the phrase “See also” is followed by a list of related Macsyma commands and topics. You can access any description by selecting its name.
- Click on the phrase `EXAMPLE(INTEGRATE)`; to execute an example of the **integrate** command. When you start the example, the program exits the hypertext help system and moves to the Macsyma Front End window, where the example will run. After each line of the example, Macsyma

prompts you to push **Enter** to move to the next line. (You may use the **space bar** in place of the **Enter** key or click on **OK**.)

If you wish to terminate the executable example for **integrate** before it ends, push the **Escape** key when the system prompts you to push the **Enter** key.³

Some commands that are described do not appear in the **MathHelp!** topic browser dialog box. Let us look at several other ways to access a description of **integrate**

1. Click on **Help** at the top of the Macsyma Front End window, then on **Index**. When the alphabetical index appears, click on the letter **I**. You will see an alphabetical list of all described topics that begin with the letter **I**. Click on **INTEGRATE** to access the description. Now return to the Macsyma Front End window by closing or minimizing the Macsyma Help window, or by clicking with your mouse on an exposed portion of the Macsyma Front End window.
2. You can also access the **integrate** description through the **Help - Search...** menu. In the Search dialog box you can type “integrate” or select it from the alphabetical list of topics.
3. In the last command line of a connected Macsyma notebook, type **describe(integrate)** and press **Enter**. The Macsyma Help window appears with the hypertext description of **integrate**. (When you type, make sure that your cursor is placed in the last input section of a connected notebook.)

Macsyma also provides context sensitive Help. That is, Macsyma can display the description of a command that appears on a command line, on an output line, or in a text section.

- At a command prompt (*Cn*), type **integrate** and press **F1**. The **integrate** description appears. This feature is very helpful when you are typing in a Macsyma command and cannot remember all of its arguments. The cursor may be anywhere within the command name to access context sensitive Help.

³ After escaping, you can continue executing the previous example, demo or batch file at the point of interruption by typing **batcon()** and pressing **Enter**.

- With the **integrate** command selected, pull down the **Help** menu; notice that the first menu item is **Help On 'Integrate.'**

2.4. Hypertext Descriptions of the Macsyma Front End

You can learn how to use features of the Macsyma Front End, which contains Macsyma's notebooks and graphics viewer, by clicking on **Help - Front End**.

2.5. Executable Examples

In the previous section you accessed an executable example of the **integrate** command:

- from inside the hypertext description of the **integrate** command.
- by clicking on the **Example** button in the **MathHelp!** topic browser.

You can also access executable examples from outside **MathHelp!** Type `example(integrate)` on the last command line in a connected Macsyma document and press **Enter**. The executable example for the **integrate** command will start. After each command in the example, move to the next line by pressing **Enter**, the **space bar**, or by clicking on **OK** in the status bar at the bottom of the Front End window. To exit the example at any time, press **Escape** or click **Cancel** in the status bar.

2.6. Function Templates

Click on **MathHelp!** at the top of your screen, then on the topic **Calculus**, then on the subtopic **Integration**, then on the Macsyma command **integrate**.

- You may click **Example** to run the executable example for the **integrate** command,⁴ without entering the hypertext system, if you choose. (Let's not do that this time, since we just viewed that example.)

⁴ Some entries on the math topic menus do not have examples or function templates. These entries are usually "option variables," which can take on different values to alter the behavior of Macsyma, but which are not themselves functions which take arguments. ("Fancy_display," for example, is an option variable with the settings **TRUE** or **FALSE**.) While some option variables have executable examples, none has a template.

- Click on the **Template** button. You will see a dialog box which lists the four arguments to the **integrate** command and indicates that the last two arguments are optional. We will use the template box to compose an integration command which performs the operation:

$$\int_a^b x^2 dx$$

by filling in “slots” in the dialog box. The cursor appears in the slot below the label “Expression to integrate.” Type x^2 in this slot. Tab to the slot below the label “Variable of Integration,” and type x . Move the cursor to the next slot, and type a ; move it to the last slot, and type b . Then click **Submit**.

Control now moves to the Macsyma Front End window, where you will see the command:

`integrate(x^2,x,a,b)`

on the current input line. When you press **Enter**, the command executes and returns the result:

$$\frac{b^3}{3} - \frac{a^3}{3}$$

2.7. Executable Demonstrations

The executable examples illustrate how to use individual Macsyma commands. The executable demonstrations illustrate how to use an entire Macsyma package or how to combine Macsyma commands to solve a problem.

- Click on the **Help** item at the top of the Macsyma Front End window, then on **Demos**. You will see a list of approximately eight topics. Macsyma includes over 200 executable demonstrations, organized by mathematical fields and application areas. A list of general demonstrations appears under each topic.
- Click on **DEMO(BEGIN)**; to start a brief, introductory demonstration. You can interact with a demonstration just as you do an example. Press **Enter** or the **Space** bar to execute the next line. Press **Escape** to terminate the demonstration.
- Click on **Help - Demos** again, then on **DEMOS_CALCULUS**. Scroll down to **Integral Calculus**. Each demonstration focuses on a distinct

aspect of Macsyma's integration capability. Click on DEMO(INTEGRATE); to see an executable demonstration of the **integrate** command that is over three times as long as the **integrate** example. (To exit, type the **Escape** key when you are prompted to continue.

DEMO(BALLISTICS); and DEMO(OSCILLATOR); are other good introductory demonstrations (They model a cannon ball in flight and a linear harmonic oscillator, respectively.) Access them from the list of demonstrations under **Help - Demos**, or from the keyboard. Let's use the keyboard method

- Type demo(ballistics) or demo(oscillator). The problem and its solution, complete with graphics, unfolds before you.

You can see a demonstration of animated graphics. Click on **File - Open** and open the Macsyma notebook **\macsyma2\demo\kleinani.mfe**. Follow the instructions to see animations of graphics.

2.8. The Interactive Primer

The most elementary introduction to interactive commands in Macsyma is the Interactive Primer. You can start the Primer by typing `primer()`. Select one of the scripts shown in the list displayed when the primer starts.

3. ENTER COMMANDS

3.1. Top Five Tips for Getting Acquainted with Macsyma.

1. The first step to getting help in Macsyma is to type a question on a (c) line followed by a ? to enter MathTips. Also, you can type F1 with the cursor positioned on a word the Help system recognizes. Don't forget about the other choices on the Help menu.
2. Input commands are not case sensitive in Macsyma.
3. **Basic Conventions:**
 - Use Enter at a (Cnn) prompt to execute your command;
 - Use Control-Enter for a new line without executing;
 - Use * for multiplication, as in 2*x (not 2 x or 2x);
 - Use : for assignment (e.g. a:4);
 - Use = for equations (e.g. solve(x^2-x=0, x) solves the equation for x);
 - Use := to define functions (e.g. f(x):=sin(x*x)+2);
 - Use parentheses for functions (e.g. sin(x) not Sin[x] or sin x.);
 - Use [] for lists/collections (e.g. [x,y,z] is a list of 3 variables);
 - Use [] for matrices (e.g. [a,b;c,d] is a 2 x 2 matrix).
4. Macsyma names its results Dnn, corresponding to Cnn.
5. Use % to refer to the previous output expression (e.g. the last Dnn). Macsyma denotes common mathematical constants as %pi for 3.14159..., %e for the base of natural logs, %i for sqrt(-1).

3.2. Enter Simple Commands

To type a command into Macsyma, you must first select the last input section⁵ in a Macsyma notebook document that is connected to a Macsyma Math Engine. Each active input section has a label (c1) or (c#) (where # is an

⁵ Each Macsyma notebook is composed of sections. The main types of sections are:

- command input sections (normally having a label (c#) on the left, where # is a positive integer)
- output sections (normally having a label (d#) or (e#) on the left), and
- text sections (containing textual matter), and "I/O" sections which result from running batch files, demonstrations and examples.

integer) on the left. You can select a section by clicking on it with your mouse. When you select the section, a (default color red) box appears around it. (If the notebook appears not to have a section, click on **Edit - Insert** to create an expression input section.)

You can type commands directly into the Macsyma Front End window in input sections.

- At a command prompt (c#), type $y:x+x$ followed by the **Enter** key. Macsyma should respond with
 (d#) $2x$
- At the next command prompt (c#+1), type y followed by the **Enter** key. This command forces the current value of the variable y to be displayed.
 Macsyma should respond with
 (d#+1) $2x$

The labels (d1), (d2), etc. appear on the output display lines and are contained in output sections of the notebook document.

You can make one command span several lines. The keyboard gesture **Shift-Enter** adds a line feed without signaling the end of the command.

- At a command prompt (c#), type
 $y : a + b + c$ **<Shift-Enter>**
 $+ d + e + f$ **<Enter>**
 Macsyma should respond with
 (d#) $f + e + d + c + b + a$

The keyboard gesture **\$ Enter** or **Control-Enter** terminates the line and suppresses the echoing of the result of the command.

- At the command prompt (c#), type `y : a + b <Enter>`
Macysma will respond with
(d#) b + a
- At the command prompt (c#+1), type `y : c + d $ <Enter>`
Macysma will not echo the result, because the command terminated with `$ Enter`.
- At the next command prompt (c#+2), type `y`
Macysma will return `d + c`, the current value of `y`.
- Repeat the exercise setting `y` to `e + f` and using `<Control-Enter>` instead of `$ Enter`.
- Check the value of `y`.

The executable examples and demonstrations in Macysma provide many samples of the proper use of Macysma commands. They are organized so that you can locate the example you want at the right time. Every command that appears on a “c-line” (command line) in a Macysma example or demonstration is one you can submit to the system as a command in interactive mode.

3.3. Edit Macysma Input Commands

Suppose you type `expand((x+y)^2)` and then realize that you meant to type `expand((x+z)^2)`.

- Type `expand((x+y)^2)`
 - Move your cursor to the left of the `y`, delete it, and type a `z`. (You can move your cursor to the point where you wish to make a change with your mouse by using the left and right arrow keys, or the **Home** and **End** keys.) When you press the **Enter** key, Macsyma will respond with

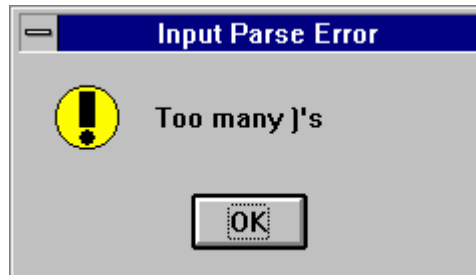
(d#) $z^2 + 2 x z + x^2$

Suppose you have submitted an expression that is not a legal command.

- Move your cursor to the current command line in a connected Macsyma document and type

`f(x)) <Enter>`

Because of the unmatched parenthesis, Macsyma displays an error message. It will look something like



- Select **OK**, and simply delete the second right parenthesis and push **Enter** to execute the command.

Macsyma will echo back

(d#) $f(x)$.

The next exercise exhibits some convenient features for re-using parts of previous commands in new commands.

- On the current command input line, type

$$y +$$
- Select the characters $(x+z)^2$ from the command line in the exercise above. Hold down the left mouse button as you drag the mouse cursor across the characters that you wish to highlight the seven characters in black.
- Press the key combination **Control+O**. The current command line should contain the string

$$y + (x+z)^2.$$
- Press **Enter**. Macsyma should echo the expression $(z + x)^2 + y$
- Press the key combination **Control+O**. This should put the previous command expression $y + (x+z)^2$ onto the current command line.

When nothing is selected (highlighted) in any previous Macsyma command input line, **Control-O** copies the most recently completed command line to the new command line. **Control-O** will copy only the highlighted (selected) section of a previous command to the new command line, so be certain you have selected the complete command.

Suppose you wish to clear the current command line completely.

- Type $f(x)+g(x)$ on the current command line.
- Press **Control-U** to delete all characters on the command line.

3.4. Make Files of Macsyma Commands

You can evaluate Macsyma commands in a text file as a batch job. Each of the example and demonstration scripts is actually a special batch job that waits for you to press the **Enter** key or the **Space** bar, or press the **OK**

button (provided you are not currently using the `batch()` command) between commands.⁶

- Type `batch("begin.dem")` and press **Enter**. You will see the demonstration “Begin” execute as a batch job, without waiting for you to type anything between lines.

You can use the batch command to execute any text file filled with Macsyma. Execution of the commands is sequential. Input sections, output sections and formatted text sections with comments are created in the Macsyma notebook.

You can also execute a text file of Macsyma commands with the `load` command. When a file is loaded, output sections are not created in the notebook and formatted text sections containing comments are not created.

To avoid uncertainty about a file's location, you can batch execute one of your command files by typing the complete file pathname as an argument to the batch command. For example, on Windows NT PCs, type⁷ `batch("c:\\dirname\\subdir\\My File.mac")`. Or use Macsyma's logical pathnames to indentify files within the Macsyma hierarchy. If your file is

⁶ The demonstrations are actually packed in one large file, `\macsyma2\demo\demo.pck`; and the examples in another, `\macsyma2\example\example.pck`. You may “unpack” any of them by typing, for example, `unpack_topic("begin.dem");`. This command creates a file `\macsyma2\demo\begin.dem` which contains the commands in the selected demonstration.

⁷ When you give a file path name to Macsyma on a DOS-Windows PC, you must change each backslash to a double backslash, because backslash in Macsyma means “take the next character literally.”

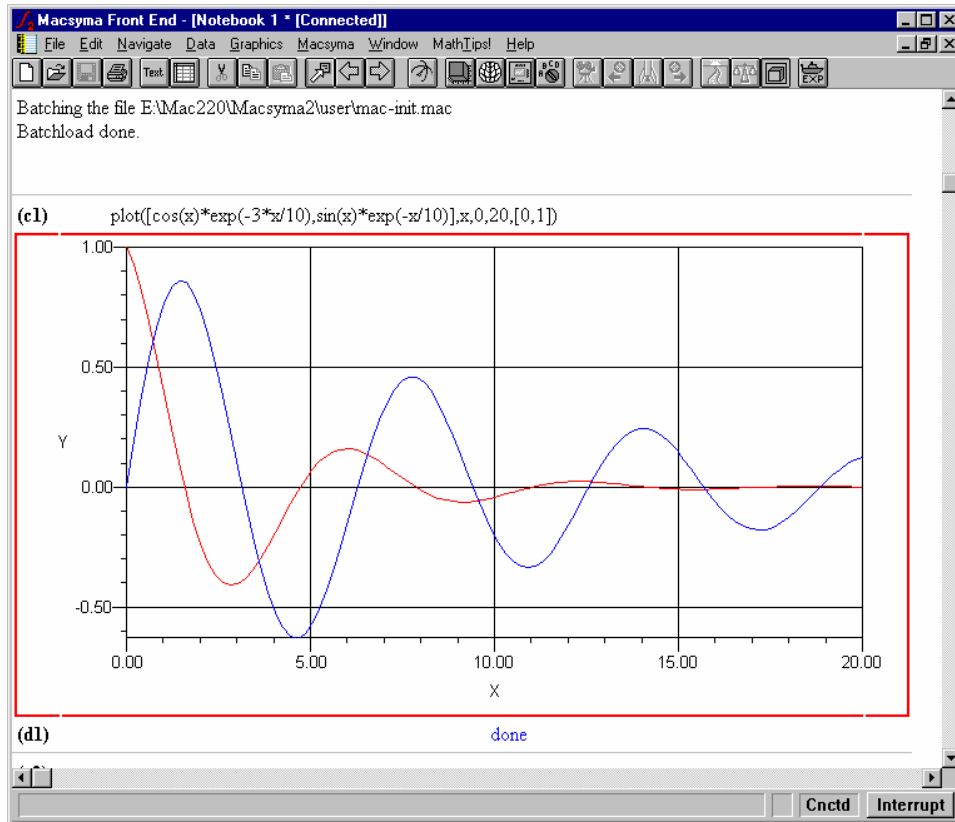
located in the `\macsyma\macsyma2\user` subdirectory, you can type `batch("macsyma:user;My File.mac");` All Macsyma commands using file names support both Windows NT/95 long file names as well as 8.3 format short file names. By convention, command files end with the extension `.mac`. The input editor looks for this extension first when seeking a file to load.

You can also turn a file of Macsyma commands into a Notebook. Select **File - Make Notebook**. All the executable commands will be placed on input lines. You can then execute them in sequence, or one at a time.

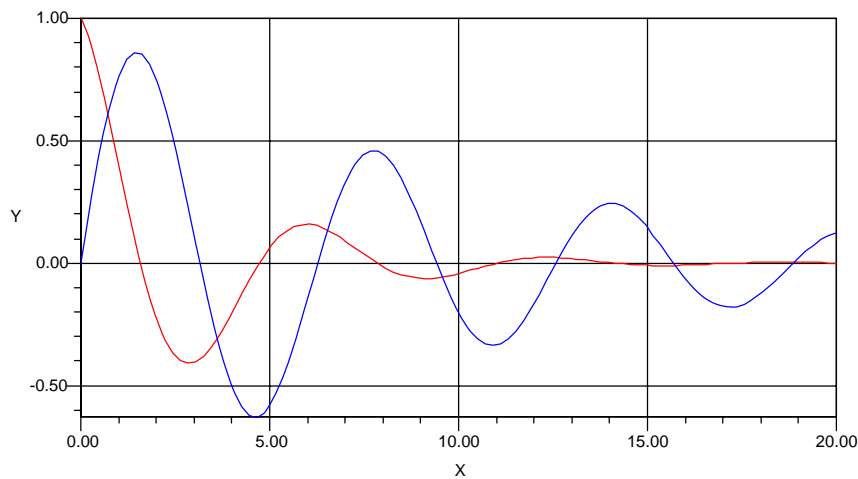
4. BUILD MACSYMA NOTEBOOKS

4.1. Create Macsyma Notebooks

When you type commands to Macsyma, you are creating a notebook that consists of input sections and output sections. Below, you will see illustrations of a graphic section, an input section and a formatted text section from a Macsyma notebook. You can place the entire notebook, or individual sections of a notebook, in another Windows application.



You can move the graphic section with standard Windows Cut and Paste techniques that produce:



You can edit individual attributes of the plot or section, such as axis labels or section brackets, and you can export the graphic section in several different graphical formats (BMP, RLE, PCX, GIF) or turn it into a Postscript or EPS file.

If you use Windows Copy and Cut and Paste, you should be aware that you are also using the Windows Clipboard. In Windows 3.X, the clipboard size is limited. Placing several sections or very complicated graphics in the clipboard can lead to Windows instability or rendering problems. Even in Windows NT or Windows 95, you may have problems rendering very large sections. Pasting notebooks with page breaks into other Windows applications may also cause problems.

Summary

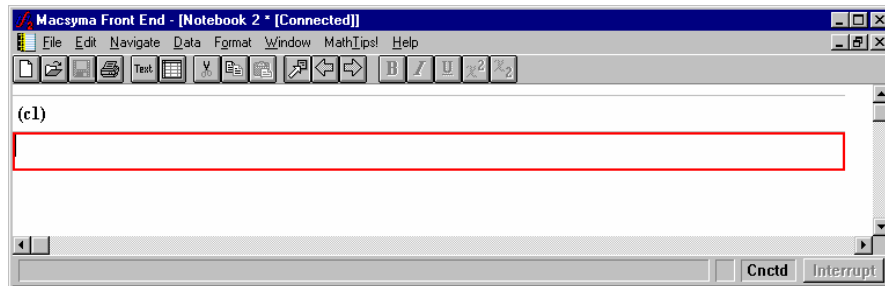
There are four basic ways to create good-looking mathematical documents with Macsyma:

1. You can create a Macsyma notebook document in the Macsyma Front End window combining formatted text, formatted mathematics (set **fancy_echo:true**) and graphics;
2. You can port Macsyma math and graphics to a word processor document using Windows Cut and Paste or Copy;
3. You can export individual graphics using the Graphics Export commands;

4. You can print individual notebook sections to a file to include in a word processor document you create later.

4.1.1. Add Text in a Macsyma Notebook

You can add a formatted text section anywhere. You can press the Text button or use Edit - Insert Text Section. In the illustration below, a formatted text section is outlined below the Expression Input Section labeled (c1).

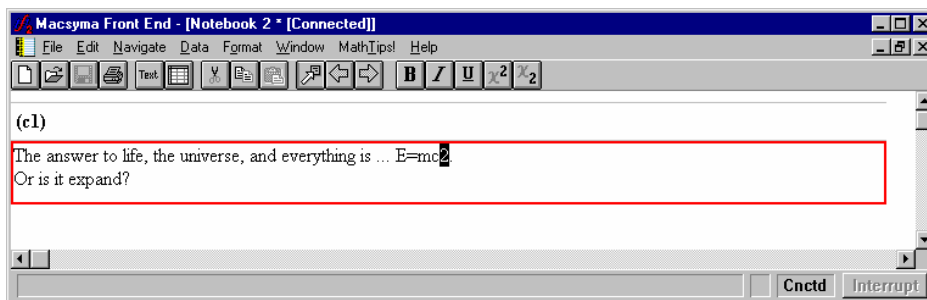


- Select the section before which you want the text to appear. Click on the command Edit - Insert Section - Formatted Text in the menu at the top of the Macsyma Front End window. An empty text section appears.
- Type

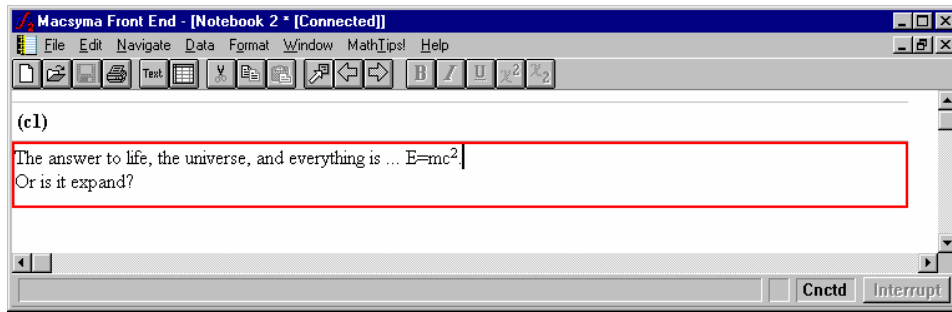
The answer to life, the universe and everything is . . . $E=mc^2$. <Shift Enter>

Or is it expand? <Enter>

Position the mouse on 2 and highlight 2. The buttons indicating font changes (bold, italic, underline) as well as superscript and subscript should light up.

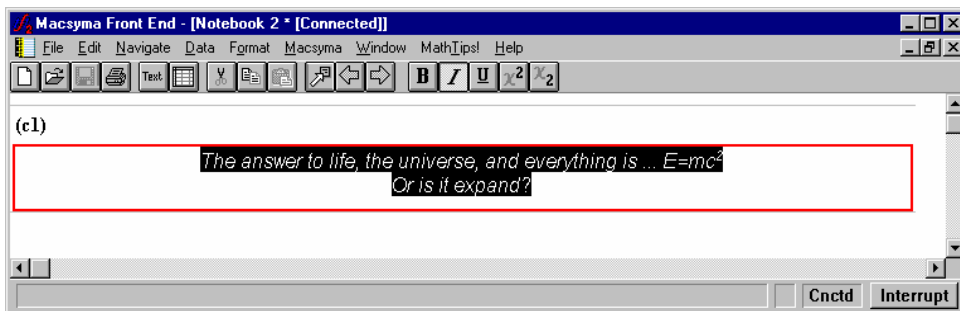


- Press the superscript 2 button, or Format - Superscript. You will see something like:

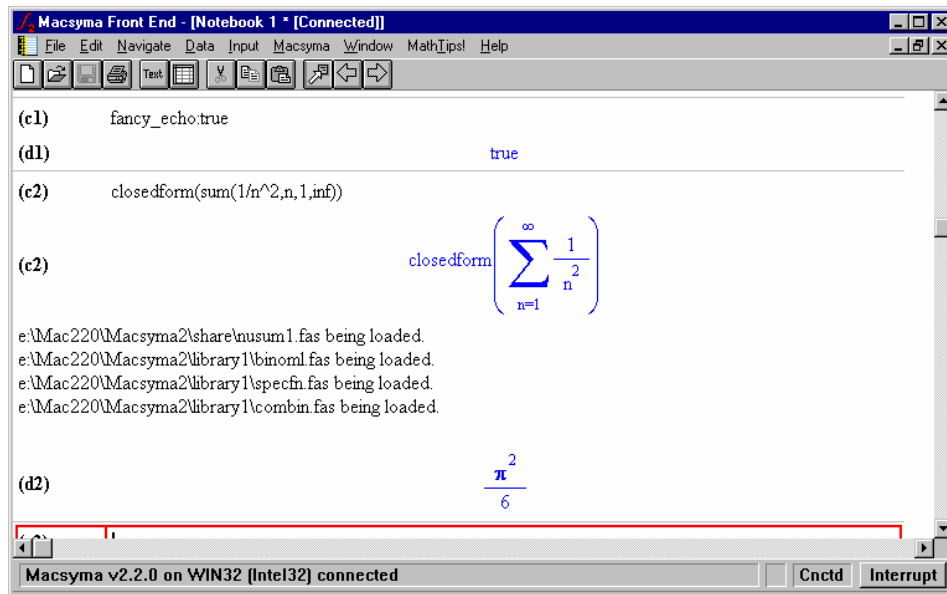


- Position the mouse inside the word **expand** and press F1. Note that text lines also have context sensitive Help.

You can introduce line feeds (with the key combination Shift-Enter) and new paragraphs (with the Enter key). You can format the text by changing its font, font size, color and making individual characters, words or phrases bold or italic. To access these controls, select a section of text and click on the menu item Format that appears at the top of the Macsyma Front End window. You can open a dialog box which contains controls for changing character formats or paragraph formats. You can change paragraph formatting to None, Center, Left or Right. The notebook, with character formatting changed to Arial Italic, and center justification looks something like:



You can use Macsyma's fancy display facility (with `fancy_echo:true`) to create formatted text sections which turn input c-lines into formatted text sections. The following notebook was created just that way:



You can create the appearance of embedding formatted mathematical expressions in text by placing an input or output section between two text sections.

4.1.2. Rearrange Sections in a Macsyma Notebook

You can rearrange the order of sections in a Macsyma notebook.

- Select any input section, output section, graphics section, text section or I/O section.
- Place the mouse cursor on the outline box of the selected section. A square document icon will appear on the tail of the mouse cursor, indicating that you have selected the thin border line.
- Holding down the left mouse button, drag the mouse to a location between two other sections where you want the selected section to

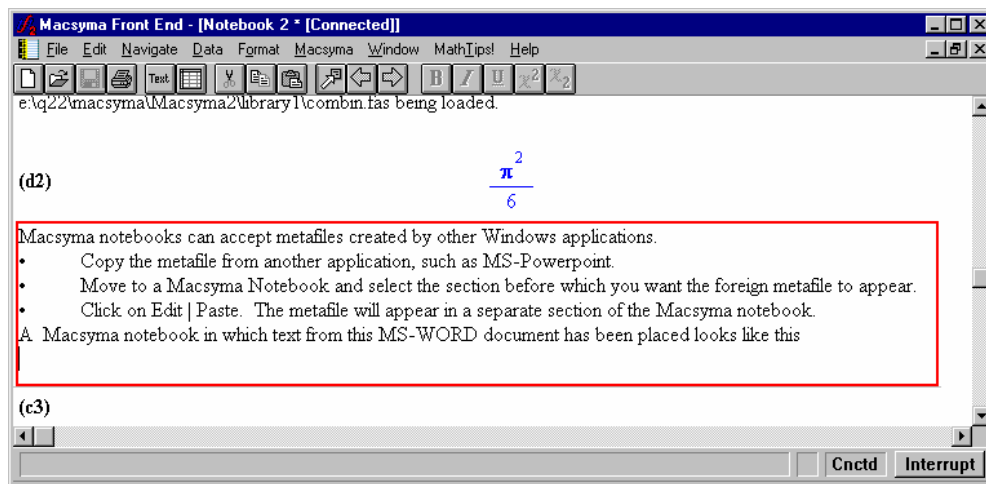
appear. Release the mouse button, and the section will move from its former location to the location specified by the mouse cursor.

4.1.3. Insert Other Graphic Objects into a Macsyma Notebook

Macsyma notebooks can accept metafiles created by other Windows applications.

- Copy the metafile from another application, such as MS-Powerpoint.
- Move to a Macsyma Notebook and select the section before which you want the foreign metafile to appear.
- Click on Edit - Paste. The metafile will appear in a separate section of the Macsyma notebook.

A Macsyma notebook in which text from this MS-WORD document has been placed looks like this:



4.1.4. Create a Notebook of Input Commands Only

To create a notebook consisting only of the input commands from a Macsyma session:

- Click on Edit - Select Input.

- Click on **Edit - Copy Section**.
- Move to an empty notebook (which you can create by clicking on **File - New**) and
- Click on **Edit - Paste Section(s)**.

4.1.5. Save and View a Macsyma Notebook

- Save the new notebook by clicking in the Macsyma Front End window on **File - Save As**; specify the file pathname as you do in all Windows applications. The pathname must end with the filename extension `.mfe`. Macsyma 2.2 registers notebooks. If you do not supply the `.mfe` extension, it will be automatically appended.
- View a notebook by clicking on **File - Open** and specifying the file pathname of the notebook you wish to open. The specified notebook will appear in the Macsyma Front End window, which supports Multiple Document Interface.

4.1.6. Re-execute Individual Commands in Place

After you have executed a Macsyma expression, it is possible to edit the expression and re-execute it.

- Select the input section you wish to edit and re-execute.
- Make changes to the expression, and press **Enter**. Macsyma will re-execute the command.

On the current command line ((c10) in our example), type :

```
(c10) ((2+y)^10)
```

- When you press **Enter**, Macsyma will return a result.
- Go back to the same command line and edit the input to read:

```
(c10) ((2+y)^10),expand
```
- Press **Enter** and note the result.
- Again, go back to the same command line and edit the input to read:

```
(c10) float((2+y)^10),expand
```
- Press **Enter** and note the result.

4.1.7. Execute and Re-execute a Macsyma Notebook

You can execute a Macsyma notebook as follows:

- View the notebook as a document in the Macsyma Front End window.
- Click on **Edit - Select - Input**. All of the input sections in the notebook will be selected.
- Click on **Edit - Reexecute**. If the notebook is connected to a Macsyma Math Engine, the engine will re-execute the input commands in the notebook. If the notebook is not connected to a Macsyma Math Engine, then Macsyma displays a dialog box that offers the choice of starting a new engine or of attaching this notebook to a math engine that is currently connected to a different notebook.

Each section in a Macsyma Notebook is marked as “replaceable” or “not replaceable” during re-execution of the preceding Input Section. When one or more Input Sections are re-executed, new output section(s) are inserted, and all replaceable sections from the current location to the next not replaceable section are removed from the notebook. The new sections are placed either immediately after the re-executed Input Section (the default) or at the end of the notebook.

By default, each Input Section, each Text Section and each Metafile Section is marked as “not replaceable.” Also, each Output Section and each Graphics Section is marked “replaceable” when they are created. Replaceable notebook sections display an extra tick mark in the upper right hand edge of the section bracket.

4.2. Create Finished Documents Inside a Word Processor

A “notebook” is a document that combines mathematical expressions, graphics and formatted text. You can create a notebook using Macsyma and any Windows-compatible word processor, such as *MS-Word*.

4.2.1. Insert Mathematical Expressions into a Word Processor Document

- Submit one or two commands to Macsyma so that you have one or more output display lines in the Macsyma Front End window to copy

to a notebook document. Open your Windows-compatible word processor and a document.

- Use Windows Metafiles when copying. Macsyma will detect when a Windows Metafile is appropriate to use.
- Using your mouse, select one or more sections from the Macsyma notebook or mark some text in an input section or a text section. To select more than one section, hold down the **Control** key while you click on the sections you are choosing.
- Click on **Edit - Copy Section** or **Edit - Copy** on the top menu bar in the Macsyma Front End window. This action copies the selected sections or text as a Windows metafile and places it in the Windows Clipboard.
- Move to your word processor window. Position the cursor where you want the selected mathematical expression(s) to appear. Click on the **Edit - Paste** at the top of the word processor window. The mathematical expression(s) will appear in your word processing document at the cursor location.

For example,

(c25) `sum(1/n^2,n,1,inf)`

(d25)
$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

was produced just this way. (You may find it to be more esthetic to turn off Section Brackets or labels with **File - Options** before you Cut and Paste)

In some Windows word processors, you can scale and crop a pasted image. (For example in *MS-Word*, click on **Format** then on **Picture**. A dialog box appears with commands for scaling and cropping the image.)

Macsyma input commands (without the line labels) can be copied as text by using the menu command **Edit - Copy** in the Macsyma Front End window and **Edit - Paste** in the word processor or other application. Macsyma output

expressions which are displayed with `fancy_display:false` can be copied and pasted as text.

4.2.2. Insert Macsyma Graphics into a Word Processor Document

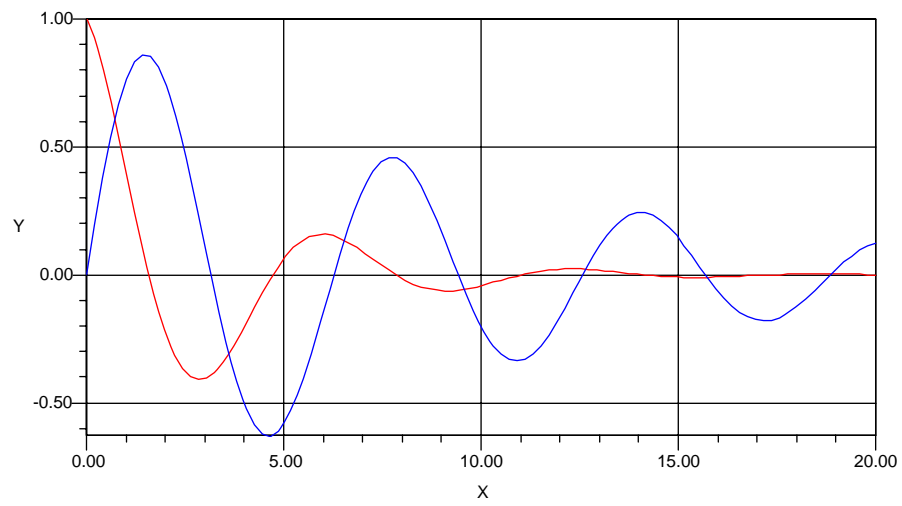
You can insert Graphics sections into a word processor document in a similar way.

1. Create a Macsyma Graphic by running `example(plot3d)` in the Macsyma Front End Window. Select the resulting plot and click with your mouse on the menu item **Edit - Copy Section**.
2. Move to your word processor window. Position the cursor where you want the graphic to appear. Click on the menu item **Edit - Paste**. The graphic will appear in your document at the location of the cursor.

4.2.3. Insert Macsyma Notebook Sections into a Word Processor Document

Suppose you want to copy a section in a Macsyma notebook into a word processor document.

- Select one or more sections from the Macsyma notebook or mark some text in an input section or a text section.
- Click on the item **Edit - Copy Section** or **Edit - Copy** on the top menu bar in the Macsyma Front End window to copy the selected sections to the Windows Clipboard.
- Move to your word processor window. Position the cursor where you want the selected text to appear. Click on the item **Edit - Paste** at the top of the word processor window. The text or graphic will appear in your word processor document at the cursor location.



The graphic above was produce just this way.

5. DO MATHEMATICS WITH MACSYMA

5.1. Expressions

You can perform calculations in Macsyma by acting on expressions. The components of a Macsyma expression can include numbers, variables, operators, and constants. Using expressions as building blocks, you can assign values to variables, create equations, and define your own functions.

5.1.1. Operators

Macsyma uses familiar operators to perform operations. Here are the operators Macsyma uses in order of priority from lowest to highest:

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
-	negation
^	exponentiation
.	non-commutative multiplication
^^	non-commutative exponentiation
!	factorial
!!	double factorial

Macsyma always uses “*” on input for multiplication. Thus, you should write $2*x$ instead of $2x$. You can optionally use “*” for display of expressions by setting the **stardisp** (Default: false) option variable. In addition to these mathematical operators, Macsyma uses “:” to assign values to variables, “=” to create equations, and “:=” to define functions. Functions use () to denote arguments, as in $\sin(x)$. Macsyma uses “[]” for lists and

collections. Macsyma also uses “[]” for matrices. See Section 5.5. Matrices, page 48.

Macsyma performs operations of equal priority from left to right. You can use parentheses to change the order of evaluation.

5.1.2. Numbers

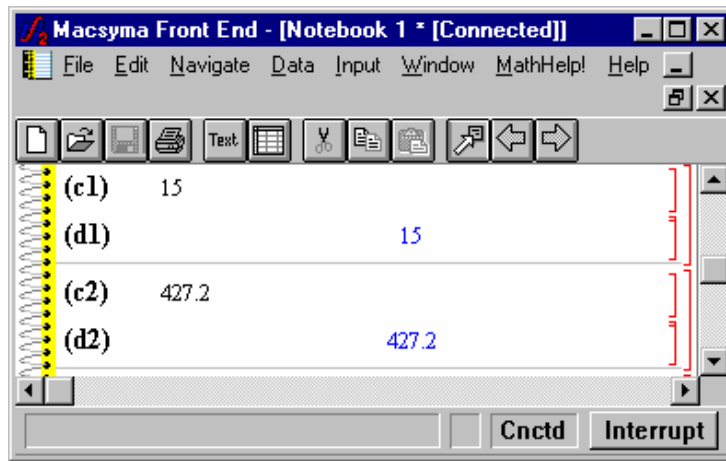
Macsyma knows about several kinds of numbers:

- **Integers** consist of a string of digits not containing a decimal point. For example, 15934. Integers can grow very large, since their size is bounded only by the total virtual address space accessible to Macsyma.
- **Rational numbers** are represented as an exact ratio of two integers. For example, $3/2$. Macsyma can represent any rational number, subject only to the memory limitations of your machine.
- **Floats** and **bigfloats** are floating-point numbers. They consist of a string of digits containing a decimal point, and are optionally followed by an e, d, or b, and an integer exponent. Examples of floats are 459.3, 83.3495e6, and 79.46d5. Examples of bigfloats are 83.3495b6 and 3957204.0b15.

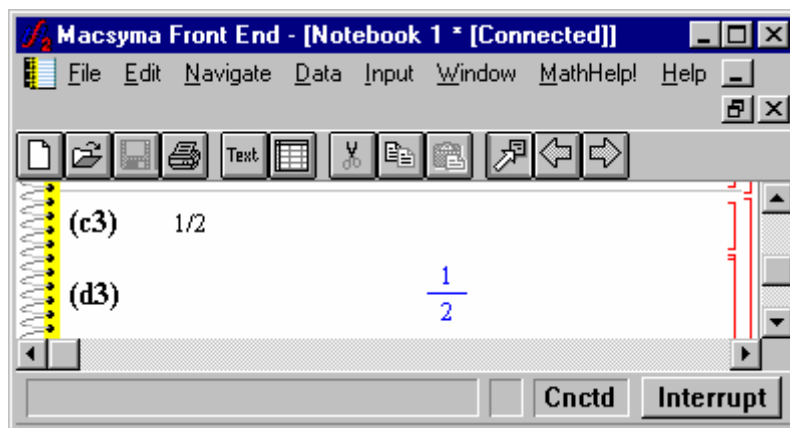
The precision of a big floating point number is limited in the Student Edition of Macsyma to 32 places. Calculations involving floating point numbers can be compromised by round-off errors.

- **Complex numbers** are written with the imaginary unit i , which in Macsyma is written %i. For example, $4i$ is written $4*\%i$.
- Negative numbers are any kind of number beginning with a minus sign. For example, -4, -17.41, -957204.0b15.

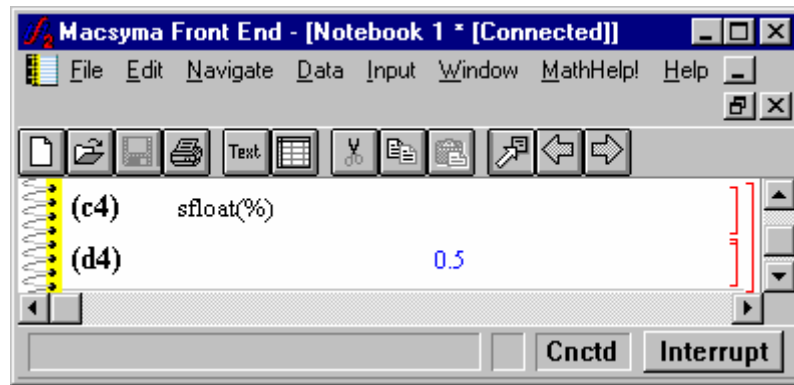
When you type a number, Macsyma simply returns it.



Macsyma preserves ratios as exact numbers rather than converting them automatically to a floating-point approximation.



At any time, however, you can use the **sfloat** or **dfloat** commands to request a single or double float approximation or the **bfloat** command to request a bigfloat approximation.

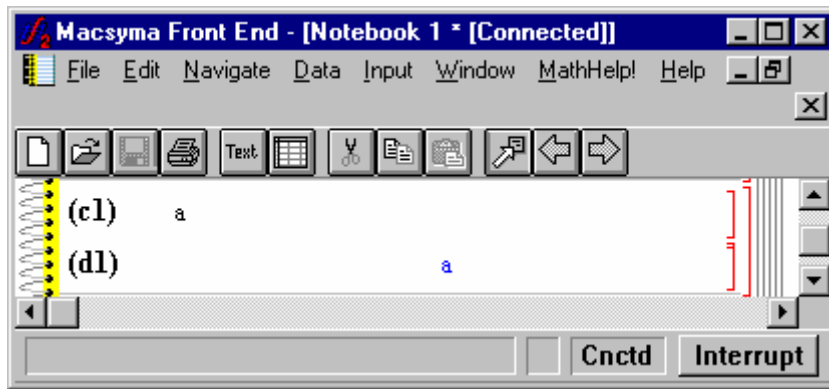


The default precision for floats is governed by each computer's hardware. If you are concerned about accuracy, use **dfloats** or **bfloats**. The bigfloat precision can be set using the Macsyma variable **bfprecision**. You should be sure to remember that “floating” a number, no matter what your precision, is an approximation. You should try to limit your use of floating point until as late in a computation as you can. Use the online Help or Tips to learn more about setting precision.

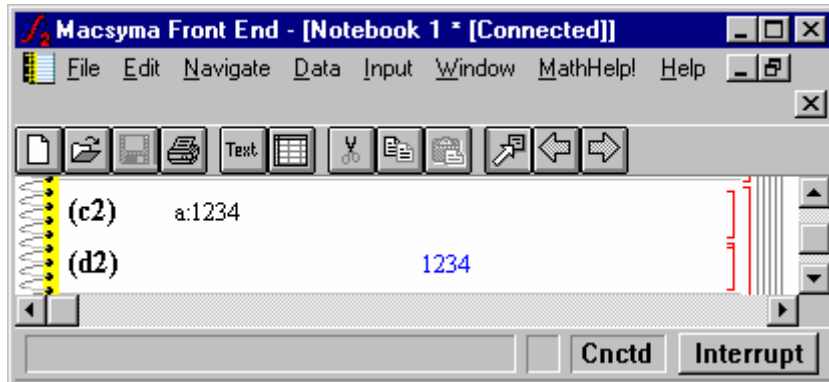
5.1.3. Variables

Variables are named quantities. You can either bind a value to a variable, or you can leave the variable unbound and treat it formally. Binding a value to a variable is called *assignment*. This section shows you how you can use both bound and unbound variables in expressions.

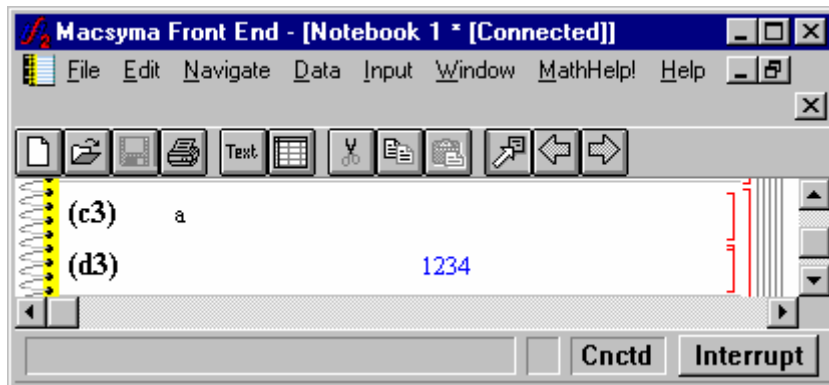
To assign a value to a variable, type the name of the variable, followed by the “:” operator, followed by the value. Do not confuse the “=” operator, which creates equations, with the assignment operator. The equation operator does not assign values to variables. When you type the name of an unbound variable, Macsyma simply returns that variable.



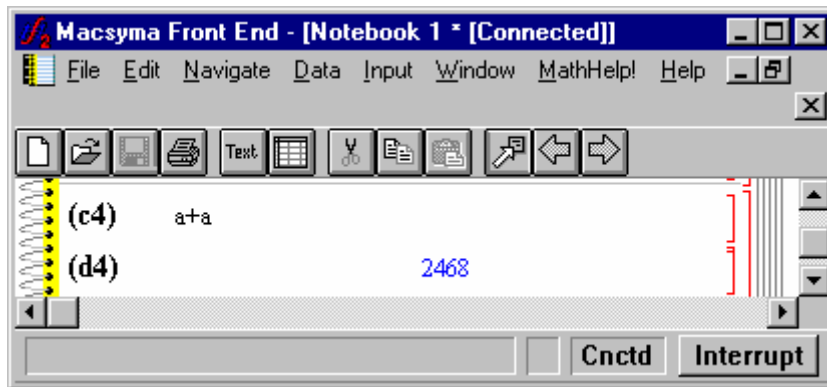
Assign the value 1234 to the variable **a**.



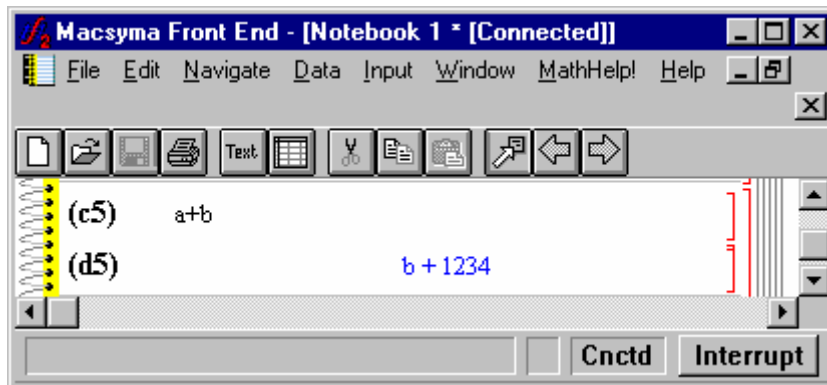
When you type the name of a bound variable, Macsyma returns its value.



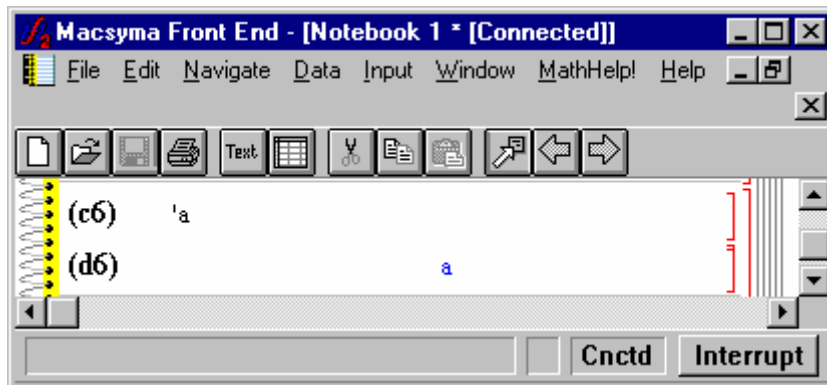
Once bound, **a** stands for 1234 in subsequent expressions.



You can add bound and unbound variables.

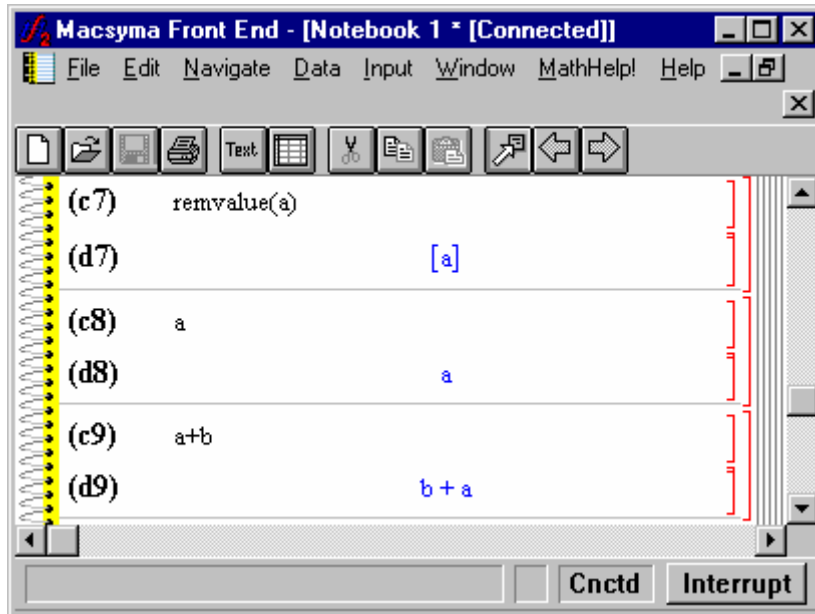


A single quote before a bound variable *suppresses* evaluation.

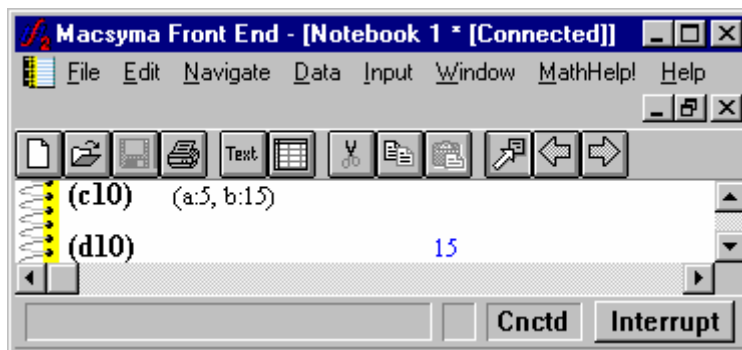


Using the variable **a** in expressions does not change its original value.

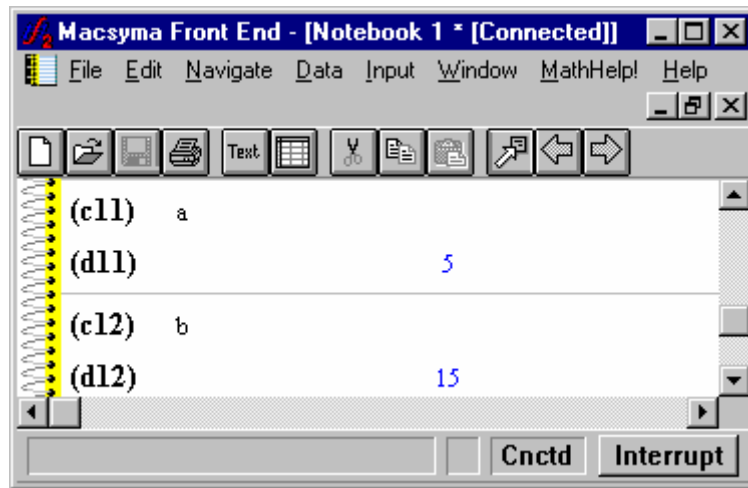
Confusing results can occur when you use a variable that you do not realize you have previously assigned. To remove a value from a variable use the **remvalue** command.



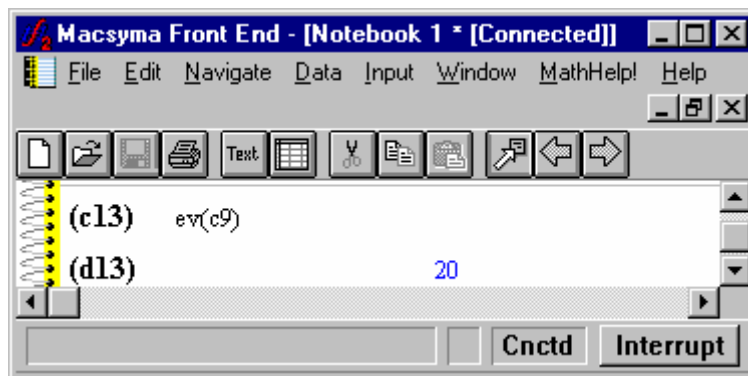
You can also create compound assignment statements by enclosing the assignments, separated by commas, in parentheses. Note that Macsyma returns the value of the last statement.



Check the values of **a** and **b**.



Use the **ev** command to re-evaluate c10; it results in the sum of the values of **a** and **b**.



Use the online Help and the Tips to find out about **ev** and **remvalue**.

5.1.4. Constants

Macsyma knows about the common mathematical constants:

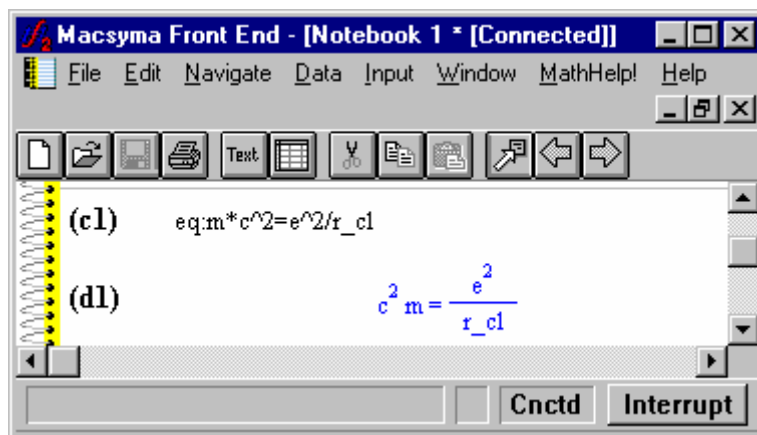
Constant	Definition
%e	Base of natural logarithms (e)
%i	The square root of -1 (i)
%pi	The transcendental constant pi (π)

inf	Real positive infinity
minf	Real negative infinity

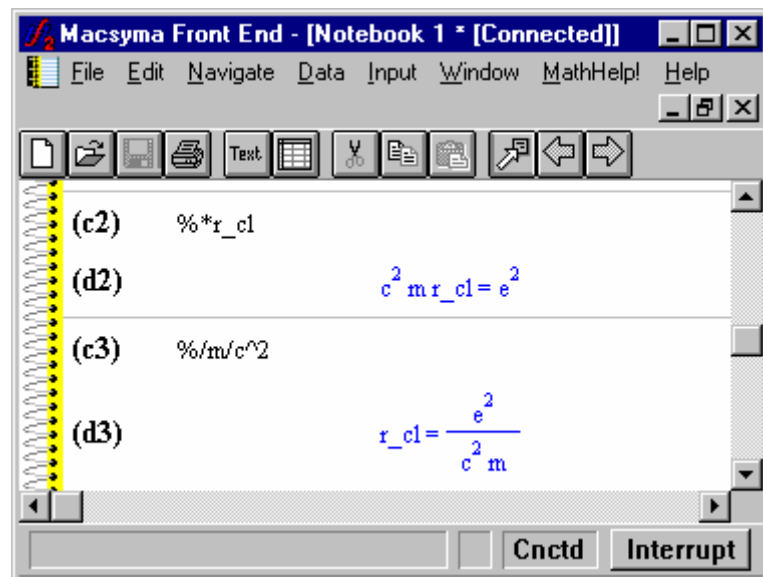
as well as some uncommon ones. Macsyma also knows about some universal physical constants, such as %%c (speed of light) and %%h (Planck's constant.) Use the online Help to find out about Physical Constants, Units, and Mathematical constants.

5.1.5. Create Equations

You use “=” to create an equation. The equal sign establishes the equality relationship between the two sides of the equation, but does not assign any value. You use “:” for value assignment, as shown in section 7.1.3. Note the use of assignment (to name the equation) and the = operator to set up the following equation:



You work with equations like any other mathematical quantities. Here we solve for *r_classical*:

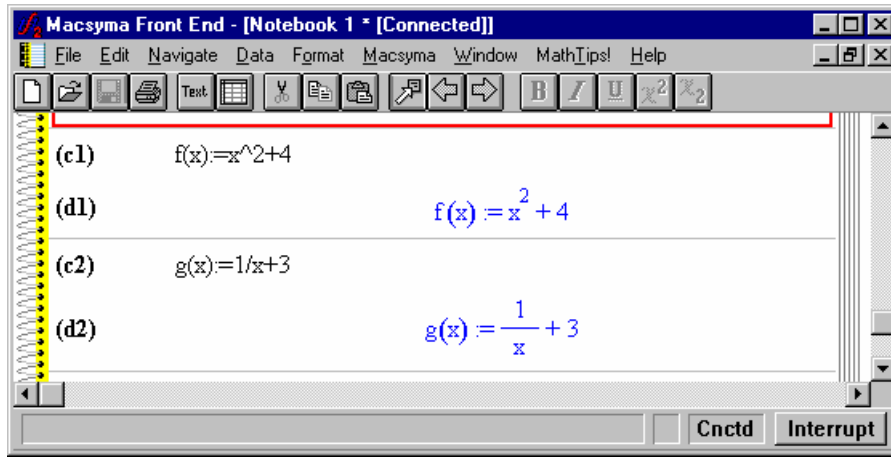


Macsyma has many powerful tools for solving equations. Use the online Help and Tips to find information and hints about Macsyma's **solve** command and related commands that can help you solve equations.

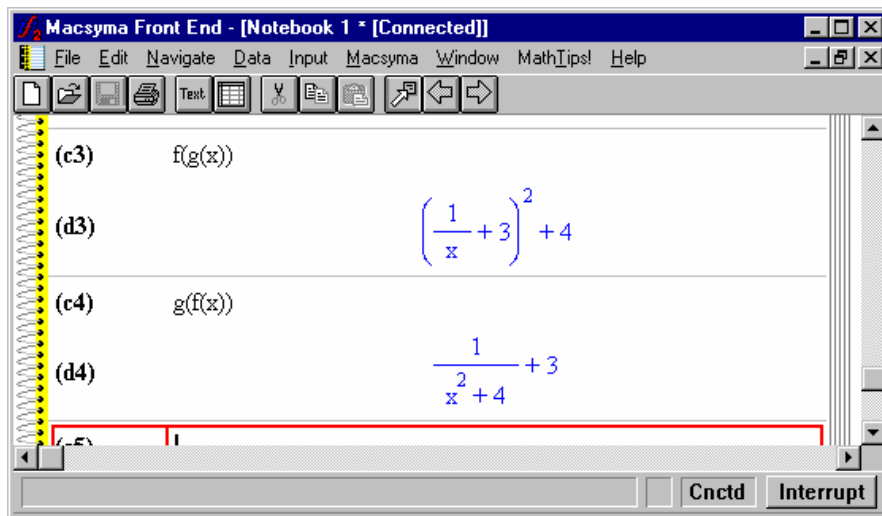
5.1.6. Define Functions

Functions are small programs that you can define to perform various calculations. To define a function, you use the syntax *function*(*arg1*, *arg2*, ..., *argn*):= *body*. Here *function* is the name of the function, *arg1* through *n* are the arguments it will operate on, and *body* is the code that describes what it will do.

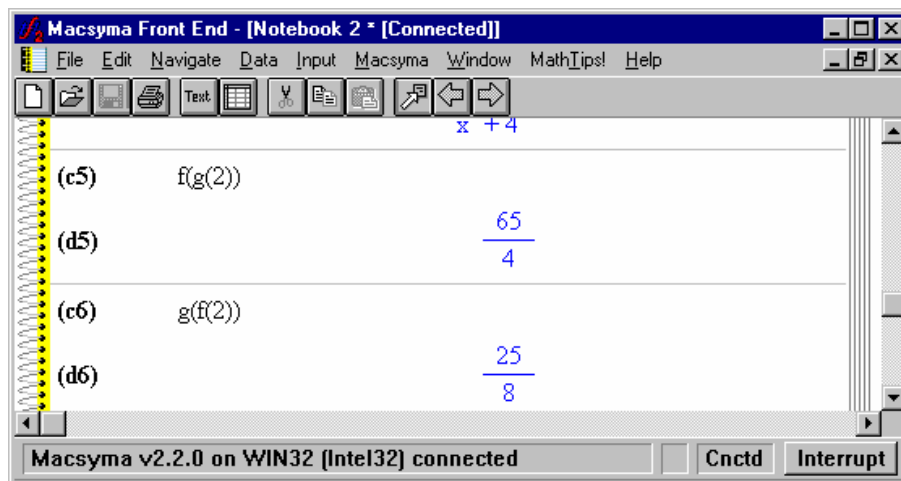
Here we define two functions:



You can compose these two functions, like this:



You can use these functions and combined functions:



Use the online Help and Tips to find out about commands for removing function definitions (**remfunction**), and ways to find out what functions you have defined. Look up **functions** and **dispfun**.

5.2. Algebra

You can use Macsyma to manipulate algebraic expressions in many ways. Macsyma can expand expressions, as you have seen with the **trigexpand** command, but it can also multiply a term or terms through a sum or equation, distribute sums over products, or do a complete partial fraction decomposition, expanding an expression in partial fractions with respect to a given main variable.

Use the online Help collection to look for Tips or function definitions for **expand**, **multthru**, **distrib**, **partfrac**, **logexpand**, and **radexpand**.

Macsyma can also simplify expressions. Use online Help to look for tips or function definitions for **ratsimp**, **radcan**, **scsimp**, **combine**, **recombine**, **xthru**, and **map**. Also check out the option variables **algebraic** and **ratfac**.

Factoring expressions is controlled by several commands. Look for **factor**, **cfactor**, and **factorsum** in the online Help.

You can substitute easily in Macsyma expressions. Look for **ev**, **subst**, and **ratsubst** in the online Help.

Macsyma can expand and simplify Trigonometric expressions. Look for **trigexpand**, **trigreduce**, and **trigsimp** in the online Help.

Macsyma can perform summations, definite and indefinite. Look for **sum**, **nusum**, **sumcontract**, and **intosum** in the online Help.

5.3. Solve Equations

You can solve equations with Macsyma in several ways. The general command is **solve**, which takes an expression and a variable as an argument, and solves the expression for that variable. If the expression is not an equation, **solve** sets the expression equal to zero in order to solve it.

Use online Help and Tips to find out about **solve**, **linsolve**, **taylor_solve**, **nroots**, **allroots**, **realroots**, and **roots** for information and examples.

5.4. Calculus

You can use Macsyma to perform the operations you have learned in Calculus, including differentiating and integrating expressions, taking limits, computing Taylor and Laurent series, solving ordinary differential equations, performing summations, and taking Laplace transforms.

HINT: It is always a good idea to check the results of integration or differentiation commands by using the opposite command. Remember that **ratsimp** and **radcan** can be your friends in helping you reduce the result Macsyma displays.

5.4.1. Differentiate Expressions

The basic command for differentiating an expression is **diff**(*expression*, *variable*). **Diff** differentiates an expression once with respect to the designated variable. Use the online Help and Tips for more information about additional arguments to **diff** that allow you to control the differentiation, take the derivative more than once, and perform more advanced operations. You should also check out **gradef**, **depends**, and **at** in the online Help for more control over differentiation.

5.4.2. Integrate Expressions

You can integrate expressions with Macsyma by indefinite integration or definite integration. Remember, floating point numbers in definite integration are subject to round off error.

Integrate is the basic command for integration. It accepts two arguments, an expression and a variable, for indefinite integration, and four arguments for definite integration (an expression, a variable, and an upper and a lower bound). The online Help has detailed information, as well as an example and demo.

Look for **integrate**, **ldefint**, **byparts**, and **changevar** for help with integration options.

For numerical integration, Macsyma has three floating point numerical methods: **quadratr**, **romberg**, and **quanc8**. See the online Help for more information.

5.4.3. Take Limits

Macsyma can find limits as a real variable approaches a given value from some direction. If you do not specify a direction, Macsyma tries to compute a bidirectional limit. The **limit** command accepts three arguments, an expression, the upper, and lower limit. It also allows you to specify a direction, so you can take a limit to some point from above or below. Use online Help to find the description of the **limit** function.

5.4.4. Taylor and Laurent Series

Macsyma has a package that can compute Laurent series. A subset of this which you may be more familiar with is Taylor series. You should look for **taylor**, **taylorinfo**, **taylor_solve**, and **tlimit** in the online help.

5.4.5. Ordinary Differential Equations

Ordinary differential equations are anything but “ordinary.” However, Macsyma provides several ways to deal with ODE’s. The command **ode** solves first and second order ordinary differential equations with dependent variables and independent variables. The syntax is **ode**(*equation*, *depar*, *indvar*).

Several Macsyma system variables may help you when you try to solve an ordinary differential equation. To evaluate a variable, type its name followed by Enter, this is true for any variables in Macsyma. Variables you may find useful include:

- **lc1** sets an initial condition for first order initial value problems.
- **lc2** sets an initial condition for second order initial value problems.

- **bc2** sets a boundary condition for second order boundary value problems.

The option variable **odetutor** and the **odelinsys** command are also useful. See their documentation in the online Help.

5.4.6. Laplace Transforms

Macsyma can compute Laplace transforms and inverse Laplace transforms. Look for **laplace**, **atvalue**, **ilt**, and **odelinsys** in the online Help.

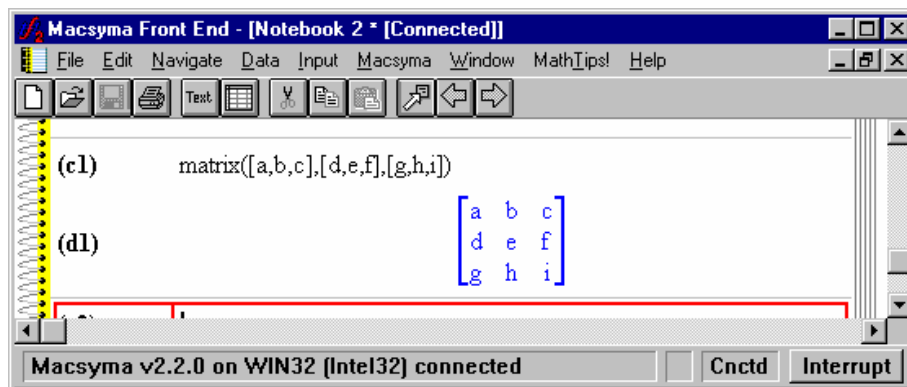
5.5. Matrices

You can direct Macsyma to create, transpose, and invert matrices. You can extract parts from matrices, add rows and columns, calculate determinants, find characteristic polynomials, and produce the echelon form of matrices.

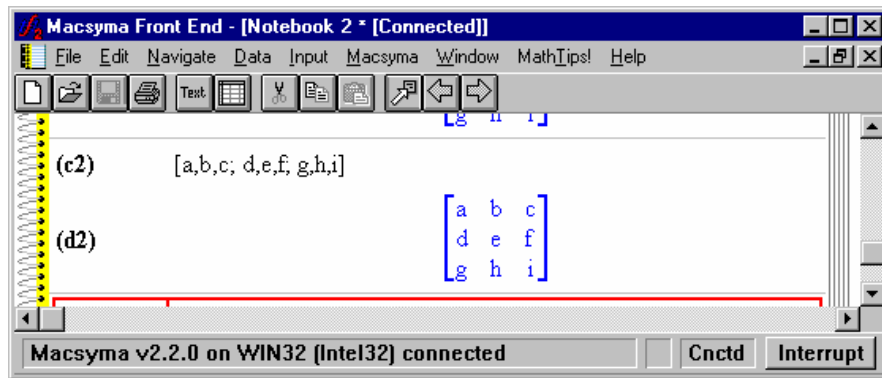
You can use the Student Edition of Macsyma to operate on a two-dimensional matrix of up to 10 by 10 elements. The large group of commands for matrix operations and option variables allow you flexible control over matrix operations.

5.5.1. Create Matrices

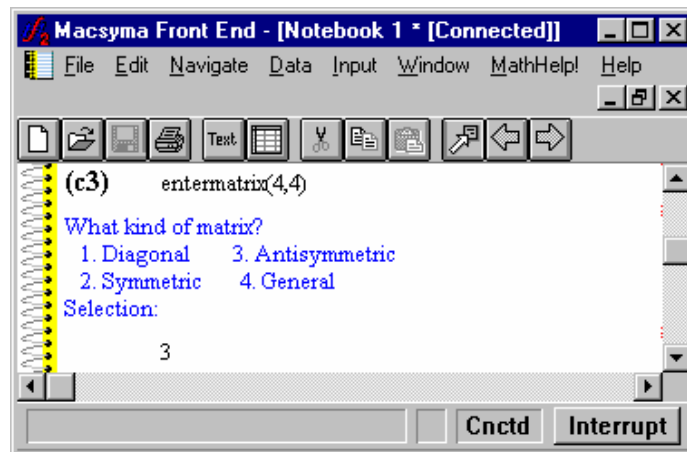
Macsyma provides commands for creating many kinds of matrices, including identity matrices. The two most general are **matrix** and **entermatrix**. **Matrix** creates a rectangular matrix from the rows of elements you specify:



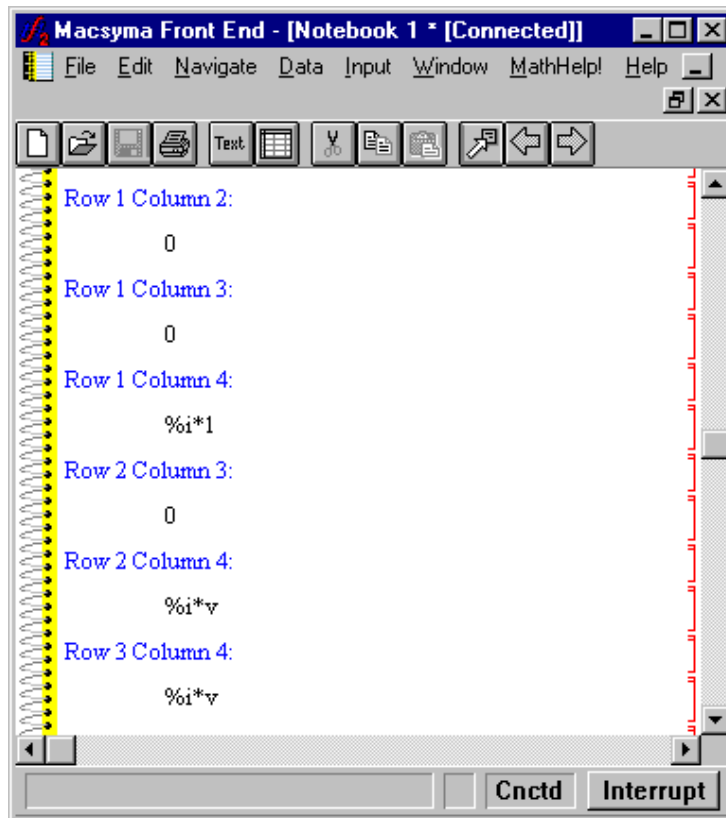
The following short-hand notation can also be used to define the same matrix:



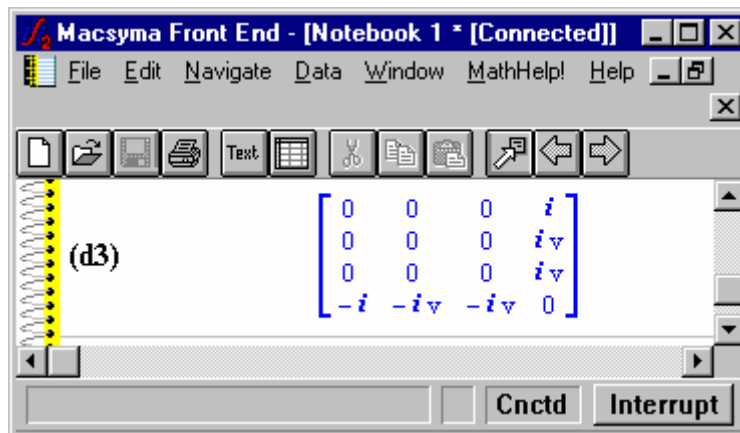
entermatrix takes the dimensions of the matrix and then prompts you for each element:



Macsyma then prompts for each element:



and the result is:



There are many other ways to create different kinds of matrices, including copying and modifying an existing matrix, and creating a matrix from an array. Use the online Help and Tips to find information about **ident**, **zeromatrix** **diagmatrix**, **coefmatrix**, **augcoefmatrix**, **copymatrix**, and **genmatrix**.

5.5.2. Operating on Matrices

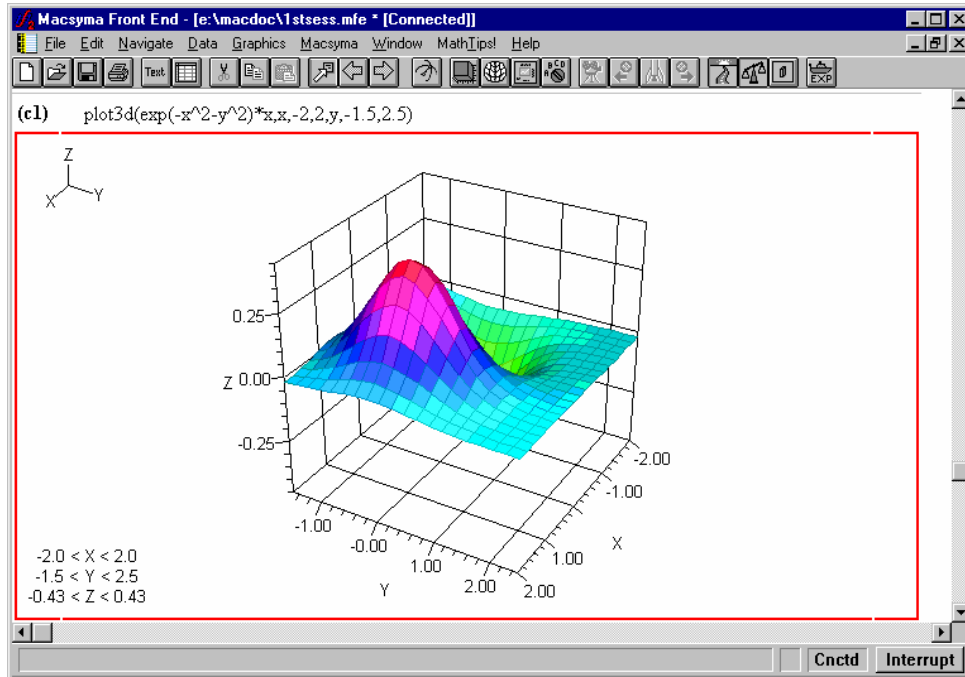
Note: most matrix commands do not actually change the input matrix. Instead, they return a new matrix, with the modification you requested. For example, you can extract a row of data in the matrix with the **row** command and perform an operation on it, but the original matrix still contains the extracted row. Similarly, the command **addrow** returns a new matrix with the specified additional row(s); it does not change the original matrix. There are similar commands for column operations, *e.g.*, **col** and **addcol**.

5.6. Display Expressions

You can direct Macsyma to rewrite your expression in many ways **using factor, combine, expand, simp**, etc. Look at the on-line Math Help on Algebra & Trigonometry as a first place to start.

6. Manipulate Macsyma Graphics

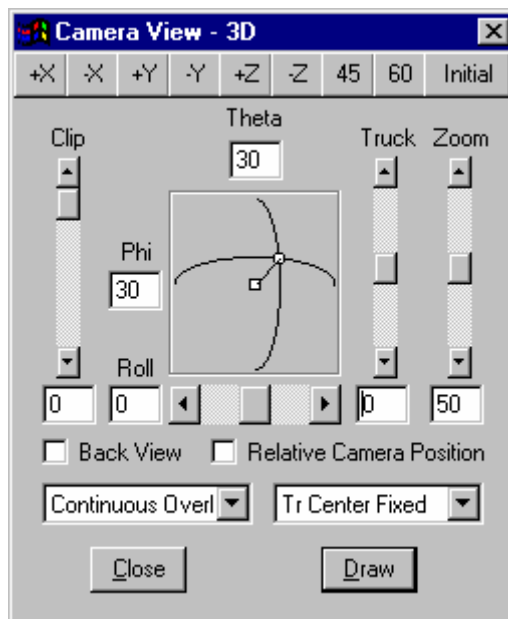
To view a 3D plot, make sure you have selected a Macsyma notebook document that is connected to a Macsyma Math Engine. Type `demo(plot3d);` When the first plot appears, exit the demo by pressing the **Escape** key. Select the plot by clicking on the frame of the plot (*not* on the object itself). A box resembling the following example will appear around the plot:



6.1. Alter the View of a Plot

- Click on the menu item **Graphics - Camera View** at the Menu Bar in the Macsyma Front End window when a graphics section is selected.⁸ A dialog box similar to the following will appear that contains a camera gimbal icon, various sliders, and other view controls:

⁸ Notice that Macsyma dynamically changes the items on this menu bar depending upon the type of section you have currently selected.

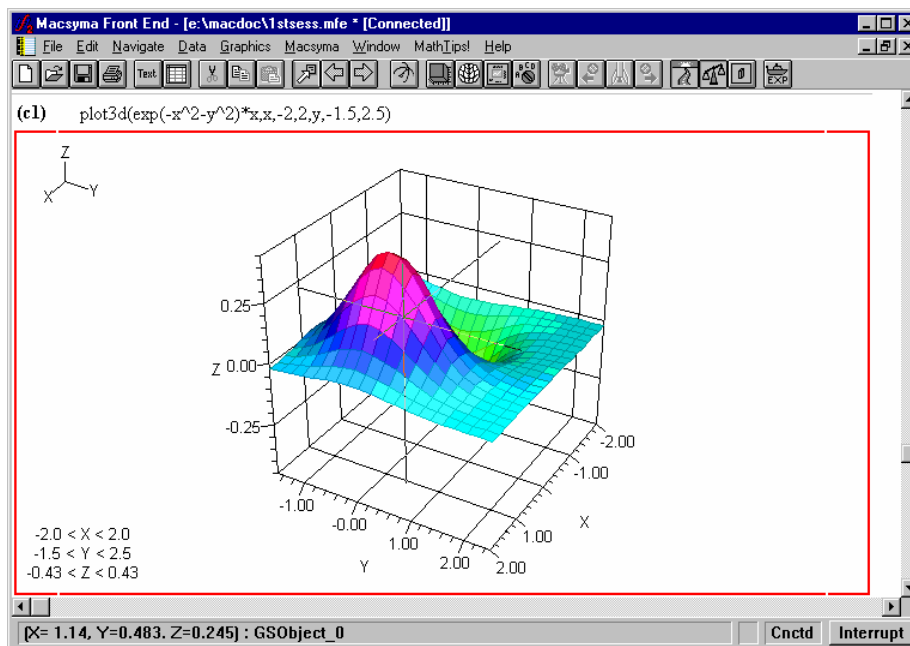


- Select the camera (the circle mounted on the gimbals) by pointing with your **left-mouse**. Move the camera to a different position. You can now view the object from the new camera position. Depending on the viewing option settings, you may need to hold the **Draw** button down to re-draw the plot.
- Move the clipping slider part way down and release it. Adjust it until you see it clipping the plotted object. Macsyma's 12 clipping planes can be adjusted from the **Camera View - Edit Attributes** dialog box, one of which is connected to the clipping slider.
- You may also alter and rotate the plot view using the keyboard arrow. Turn the **Scroll Lock** key to On and use the arrow keys to translate the plot. Use the **Shift** key with the arrow keys to increase the size of your moves.

6.2. Query a Plot for Coordinates of a Point

- Click on the plotted object. The three numbers that appear in the status bar at the bottom of the notebook document are the 3D coordinates of the

point on which you clicked. You will also see some faint lines that draw reference cursor lines through the point you selected and connect these reference lines to the coordinate axes. The resulting plot will look like the following:



- Click on the plot again and hold the left mouse button down while you drag the mouse across the plotted object. The x, y, z coordinates in the status bar change continuously as you move the mouse.
- Click again in the graphics section (but not on the plot) with left-mouse to turn off the reference cursor lines. Clicking with right-mouse will offer a menu of other options.

6.3. GRAPHICS DEMOS AND EXAMPLES

The online Help system has many examples of 2D and 3D plots you can make with Macsyma.

6.4. MACSYMA GRAPHICS STYLES

Macsyma supports *graphics styles*. A graphic style is a collection of attributes relating to the way a graphic is displayed. See the *Macsyma Scientific Notebook Interface Reference Manual* for a description of notebook attributes

and decorations. Potentially, any such attribute can be a member of a given style.

7. MEET THE DATAVIEWER™ AND MFE_DATA PACKAGE

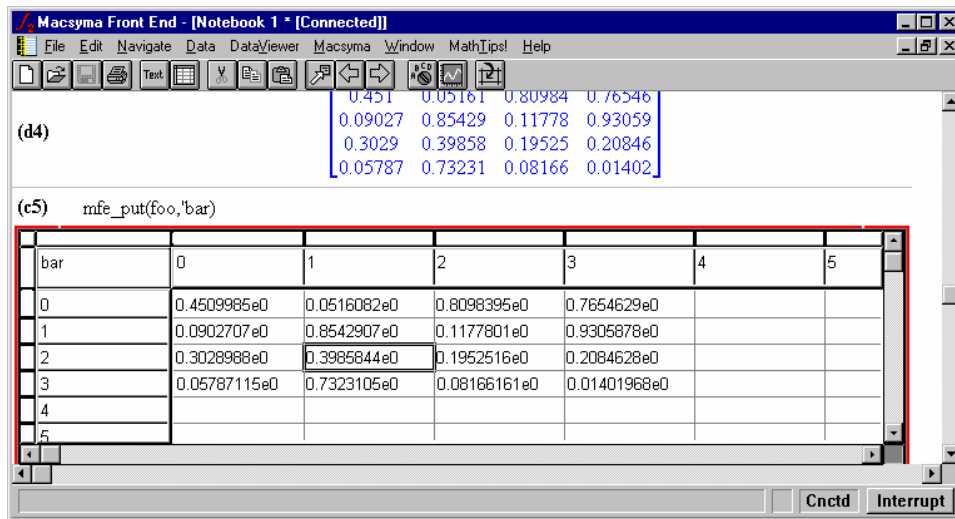
The DataViewer™ is a Macsyma Front End facility that enables you to manipulate numerical data. You can see data from a Macsyma Front End variable. You must transfer numerical data in Macsyma to an MFE variable before you can put in a DataViewer. You can also import data from a file to an MFE variable. The two basic commands for transferring data between the Macsyma Math Engine and the MFE Math Engine are **mfe_put** and **mfe_get**. The MFE_DATA package also provides new Macsyma commands for moving data back and forth between the Macsyma math engine and the MFE math engine and enables Macsyma to perform certain operations in MFE. If you try to create a new DataViewer section, a dialog will appear asking you for a math expression, which should be the name of an existing MFE math variable. See the *Scientific Notebook User Interface Reference Manual* for more information.

First, make sure that you load the MFE_DATA package by giving the Macsyma command `load(mfe_data)`. You can import data from the file system from the menu bar by selecting **Data - Import**. You can create a DataViewer section by pressing the DataViewer button on the button bar. A dialog will ask for the name of an MFE math expression, which is an MFE variable. If you want to view a Macsyma math object instead, use **mfe_put** to create an MFE variable. For example, you can make a Macsyma object, say a 4 X 4 matrix of single precision random numbers, by saying

```
foo : sfloat(mat_rand(4,4))
mfe_put(foo,bar)
```

and then place bar in the DataViewer by clicking on the DataViewer button and selecting the MFE math expression bar.

The DataViewer section in your notebook should look something like:



Note the single precision format of the numbers in the DataViewer. By default, the MFE Math Engine uses double precision numbers. When you use **sfloat** to create the matrix, the Macsyma Math Engine uses single-precision numbers, denoted with *e*-format, and the MFE Math Engine treats **bar** as a single precision array.

You can now update the MFE variable **bar** by typing new numbers. Here, the [2, 1] cell of **bar** is highlighted. Note that indices of MFE 2D arrays start at 0, while Macsyma matrices have indices that start at 1. If you change the [2, 1] cell to, say, -4, and then retrieve **bar** to Macsyma, you get a new matrix **mat1**. Your screen should look something like:

The screenshot shows the Macsyma Front End interface. At the top, there is a menu bar with options: File, Edit, Navigate, Data, DataViewer, Macsyma, Window, MathTips!, and Help. Below the menu bar is a toolbar with various icons. The main window displays a table with 6 columns and 6 rows. The first column is labeled 'bar' and contains indices 0 through 5. The other columns contain numerical values in scientific notation. The value at row 3, column 2 is highlighted in blue and is -4. Below the table, the command prompt shows the command (d5) done and (c6) mfe_get(bar). Below that, the execution time is shown as Time= 113 msecs. Finally, the result (d6) is displayed as a 4x4 matrix of numerical values.

bar	0	1	2	3	4	5
0	0.4509985e0	0.0516082e0	0.8098395e0	0.7654629e0		
1	0.0902707e0	0.8542907e0	0.1177801e0	0.9305878e0		
2	0.3028988e0	-4. e0	0.1952516e0	0.2084628e0		
3	0.05787115e0	0.7323105e0	0.08166161e0	0.01401968e0		
4						
5						

(d5) done

(c6) mfe_get(bar)
Time= 113 msecs

(d6)

0.451	0.05161	0.80984	0.76546
0.09027	0.85429	0.11778	0.93059
0.3029	-4.0	0.19525	0.20846
0.05787	0.73231	0.08166	0.01402

Cnctd Interrupt

Note that the [3,2] element of the Macsyma matrix D6 is -4. You can export the MFE variable **bar** to a file by selecting the Menu item **Edit - Export**. If you choose to export **bar** to a CSV file format with default settings, you get a file that looks like:

```
0.4509985,0.05160820,0.8098395,0.7654629
0.09027070,0.8542907,0.1177801,0.9305878
0.3028988,-4.000000,0.1952516,0.2084628
0.05787115,0.7323105,0.08166161,0.01401968
```

You can choose the separator or select fixed format when exporting. Similarly, you can use the Menu item **Edit - Import** to import a data file in CSV or fixed-field format. You could prepare such a file with a spreadsheet, like Excel or Lotus 123, or use a spreadsheet to read a file exported from Macsyma.

See `USAGE(MFE_DATA)`; for more information, and do `DEMO(MFE_DATA)`; for a demonstration.

7.1. The MFE Math Engine and Variable Names

There is a separate math engine inside the Macsyma Front End that operates on Macsyma Front End variables, which are separate from Macsyma Math Engine variables. You can access MFE variables from a Macsyma Math Engine by using the `MFE` function as shown above.

By default, the MFE Math Engine treats floating point numbers as double precision. To indicate that a number is single precision, use *e*-format. For example, 1.0 is treated as double precision 1.d0, while 1.e0 is single precision.

MFE variables are local to each notebook, while Macsyma variables are local to each Macsyma math engine.

The two basic Macsyma commands to exchange data between MFE and Macsyma are: `mfe_put` and `mfe_get`. You can move Macsyma variables to MFE variables with the Macsyma command `mfe_put(mac_name, mfe_name)`. Here, `mac_name` is the name of a Macsyma variable, and `mfe_name` is the name of an MFE variable. You can retrieve MFE variables to Macsyma with the Macsyma command `mfe_get(mfe_name)`.

`Mac_name` must evaluate to:

- a floating point number (*e.g.*, 1.0, 1.e-4, -1.d3); or
- an integer (*e.g.*, 4, -5 in the range -2^{29} to $2^{29}-1$); or
- a list of such numbers (*e.g.*, [1,2,4.d0]); or
- an array of such numbers; or
- a matrix of such numbers; or
- a Macsyma string (*e.g.*, "AbCdEfG").

Symbolic expressions, atoms, rational numbers, big integers or bigfloats are not valid types for MFE variables. In particular, complex numbers are not valid.

`mfe_get(mfe_name)` will return the value of the MFE variable `mfe_name` as a number, a list, a string, or a matrix.

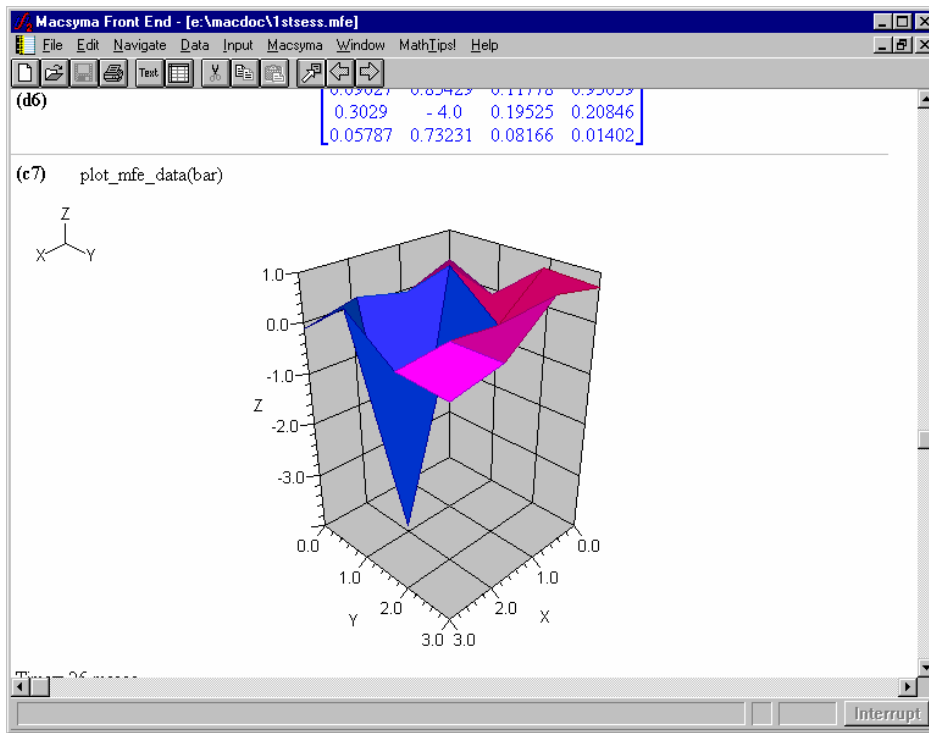
The Macsyma package `MFE_DATA` has several Macsyma commands to make it easy to work with MFE variables. Do `DEMO(MFE_DATA)`; for a demonstration. The source code for `MFE_DATA` is in the **macsyma:share;** directory.

7.2. Graphics Sections and MFE Variables

You can make a graphics section from an MFE variable with the menu item **Data - Graph**. A graphics section based on an MFE variable is like a graphics section based on a Macsyma variable or expression in that it shares the graphics decorations and attributes. However, the graphic section differs in an important way: it allows automatic updating of the graphic when the MFE variable changes. You need to recalculate the entire plot if you change, say, one number in a Macsyma list. With an MFE variable, for example, you can change one number in the DataViewer, and the associated graphic section updates automatically.

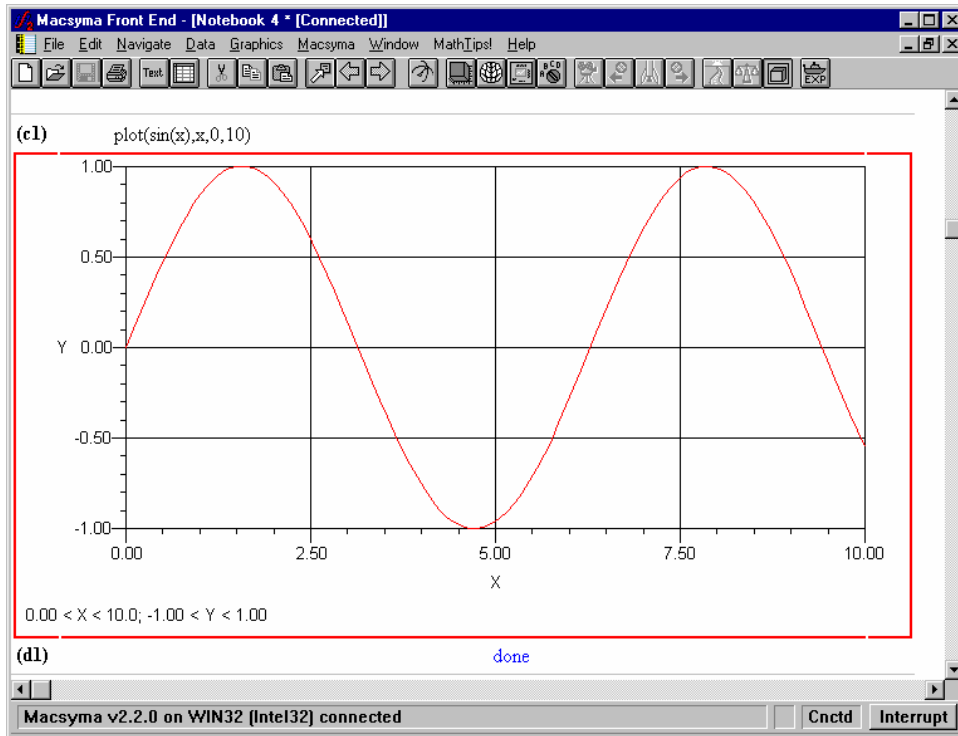
You can make a graphics section from an MFE variable `mfe_name` with the **Data - Graph** menu item, from the Graph button when it is in the Data Viewer, or with the Macsyma command `plot_mfe_data(mfe_name)`.

By default, if `mfe_name` is a 1D MFE variable, you get a plot of `mfe_name` vs. its integer index as seen in the DataViewer. If `mfe_name` is a 2D variable, you get a surface plot of `mfe_name` vs. its 2D integer indices. Try the command `plot_mfe_data(bar)`. You should get a graphic section that looks something like:



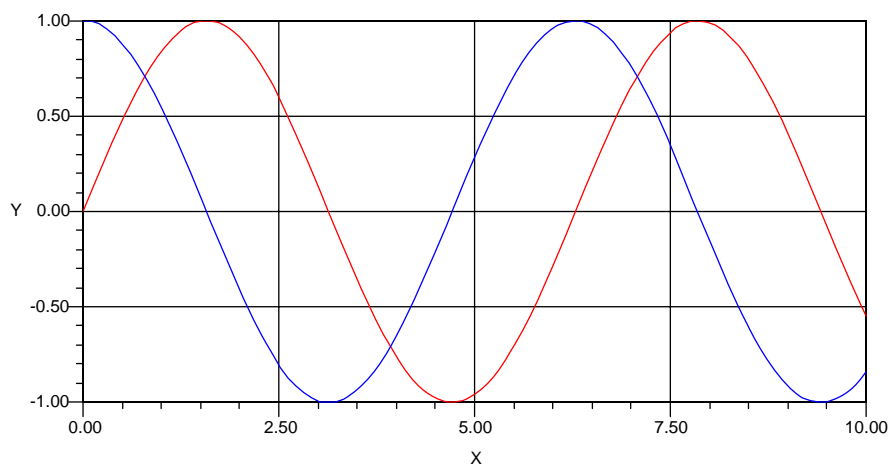
You can plot more than one MFE variable at a time by using the menu item **Data - Add Data to Graph**.

You can save the values of the coordinates of the plot of a Macsyma expression to an MFE variable by selecting the Menu item **Graphics - Assign Data to Variable**. For example, you can create a graphic from a Macsyma expression by saying `plot(sin(x),x,0,10)`. You can then select **Graphics - Assign Data to variable** and assign the data to an MFE array variable called *sine*. If you now look at *sine* in a DataViewer, you will see the coordinates of the plot. It should look something like:



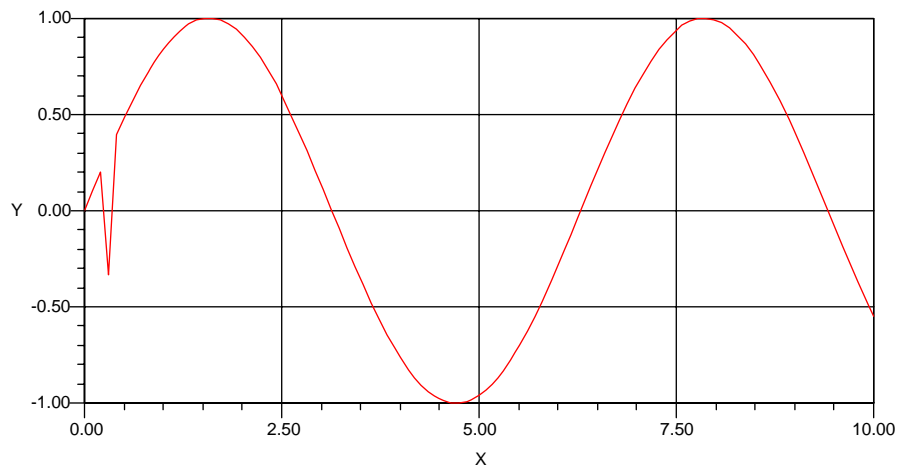
	sine	1	2	3
0.e0	0.e0			
0.1010101e0	0.1008384e0			
0.2020202e0	0.2006488e0			
0.3030303e0	0.2984138e0			
0.4040403e0	0.3931365e0			
0.5050504e0	0.4838516e0			
0.6060605e0	0.569634e0			
0.7070704e0	0.6496091e0			
0.8080807e0	0.7229624e0			
0.9090905e0	0.7889452e0			
1.010101e0	0.8468854e0			
1.111111e0	0.8961918e0			
1.212121e0	0.9363625e0			
1.313131e0	0.9669874e0			
1.414141e0	0.9877546e0			
1.515151e0	0.9984522e0			
1.616161e0	0.998971e0			

	sine_DPr2D_0_X	sine_DPr2D_0	sine_DPr2D_1	3
0	0.e0	0.e0	1.e0	
1	0.1010101e0	0.1008384e0	0.9949026e0	
2	0.2020202e0	0.2006488e0	0.9796631e0	
3	0.3030303e0	0.2984138e0	0.9544365e0	
4	0.4040403e0	0.3931365e0	0.9194801e0	
5	0.5050504e0	0.4838516e0	0.87515e0	
6	0.6060605e0	0.569634e0	0.8218985e0	
7	0.7070704e0	0.6496091e0	0.7602682e0	
8	0.8080807e0	0.7229624e0	0.6908872e0	
9	0.9090905e0	0.7889452e0	0.6144633e0	
10	1.010101e0	0.8468854e0	0.5317752e0	
11	1.111111e0	0.8961918e0	0.4436663e0	
12	1.212121e0	0.9363625e0	0.3510342e0	
13	1.313131e0	0.9669874e0	0.2548237e0	
14	1.414141e0	0.9877546e0	0.1560156e0	
15	1.515151e0	0.9984522e0	0.05561659e0	



0.00 < X < 10.0; -1.00 < Y < 1.0

	sine_DPr2D_0_X	sine_DPr2D_0	sine_DPr2D_1	3
0	0.e0	0.e0	1.e0	
1	0.1010101e0	0.1008384e0	0.9949026e0	
2	0.2020202e0	0.2006488e0	0.9796631e0	
3	0.3030303e0	-0.33e0	0.9544365e0	
4	0.4040403e0	0.3931365e0	0.9194801e0	
5	0.5050504e0	0.4838516e0	0.87515e0	
6	0.6060605e0	0.569634e0	0.8218985e0	
7	0.7070704e0	0.6496091e0	0.7602682e0	
8	0.8080807e0	0.7229624e0	0.6908872e0	
9	0.9090905e0	0.7889452e0	0.6144633e0	
10	1.010101e0	0.8468854e0	0.5317752e0	
11	1.111111e0	0.8961918e0	0.4436663e0	
12	1.212121e0	0.9363625e0	0.3510342e0	
13	1.313131e0	0.9669874e0	0.2548237e0	
14	1.414141e0	0.9877546e0	0.1560156e0	
15	1.515151e0	0.9984522e0	0.05561659e0	



0.00 < X < 10.0; -1.00 < Y < 1.0

Replotting just picks up the first one.

Note that the Macsyma Math Engine uses single precision floats for graphics coordinates.

You cannot assign the data from the plot of an MFE variable because it is already an MFE variable.

You can also take a 2D or 3D graphic of a Macsyma variable section and click on it. One of the choices offered is **Assign Data to Variable**. You can assign the graphical data to an MFE variable and view it in a DataViewer. You can replot the MFE variable and select the menu item **Graphics - Add Data to Plot** to add more data to the plot.

In contrast to the dynamic behavior of the plot of an MFE variable, the plot of a Macsyma variable will not automatically update if the data changes. Also, you cannot dynamically add new data to the plot of a Macsyma variable.

See `USAGE(MFE_DATA)`; for more information, and do `DEMO(MFE_DATA)`; for a demonstration.

8. ANSWERS TO SOME COMMON QUESTIONS

8.1. How Do I Open a Macsyma Notebook?

You can open a new Macsyma notebook (a file with extension .mfe) in several ways. First, you can click on the Macsyma Icon. Next, you can click on the MFE icon, and use **File - New** to create a new notebook. Then use **Edit - Insert** to create an expression input section.

Macsyma 2.2 for Windows NT/95 also supports Drag and Drop of Macsyma notebooks. When you select a Macsyma notebook, you can drag it with the mouse and drop it on the Macsyma 2.2 or the MFE icon to open the notebook with the Macsyma Front End. You can also double-click with on the notebook to start the Macsyma Front End and open the notebook.

8.2. How Do I Save a Macsyma Notebook?

You can save a Macsyma notebook in several ways.

Use **File - Save as** to save the notebook with a default extension .mfe so that it can be opened with the Macsyma Front End later.

Save the notebook as a .mac file. The .mac extension creates an ASCII text file that consists of the input sections and text sections enclosed in Macsyma comments. The .mac file can be used as a Macsyma batch file.

Save the notebook as an ASCII text file with extension .txt. A .txt file contains input lines, labeled as c-lines, output lines labeled as d-lines and formatted text sections. You cannot turn a .txt file into a batch file.

8.3. Is Macsyma Case Sensitive?

Most Macsyma commands are not case-sensitive. You may type commands in upper case, lower case, or mixed case and still be understood. You do not have to use the same case as is used in the command examples in this Installation Guide. (You can choose the case Macsyma uses for its output display. See **display_case** in the Macsyma Reference Manual .)

8.4. What Do I Press After Entering Commands?

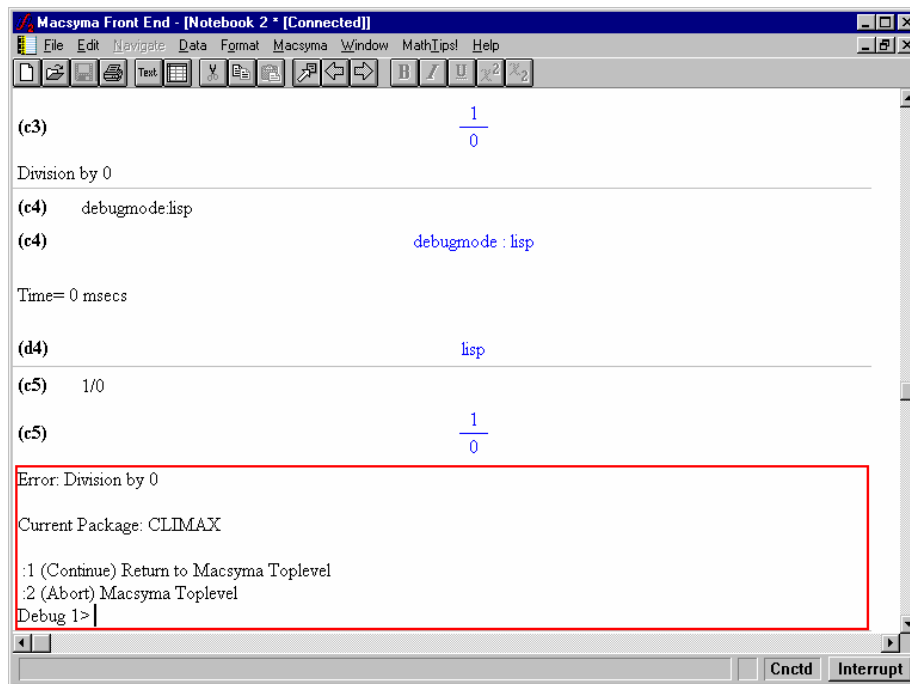
Macsyma commands end with the **Enter** key or dollar sign (\$) followed by **Enter**.⁹ The dollar sign suppresses display of the output resulting from executing the command line. You may also terminate command lines with a semicolon (;) followed by **Enter**. The semicolon is not necessary; it has the same effect as terminating the command line with **Enter**. Typing **Ctrl-Enter** has the same effect as typing \$.

In the debugger you may encounter Lisp commands. Most Lisp expressions do not require pressing **Enter** for confirmation. Lisp activates as soon as you type the right-hand parenthesis which terminates the expression.

⁹ Line feed is accomplished with the key combination **Shift-Enter**.

8.5. What Do I Do If There Is an Error Or Break?

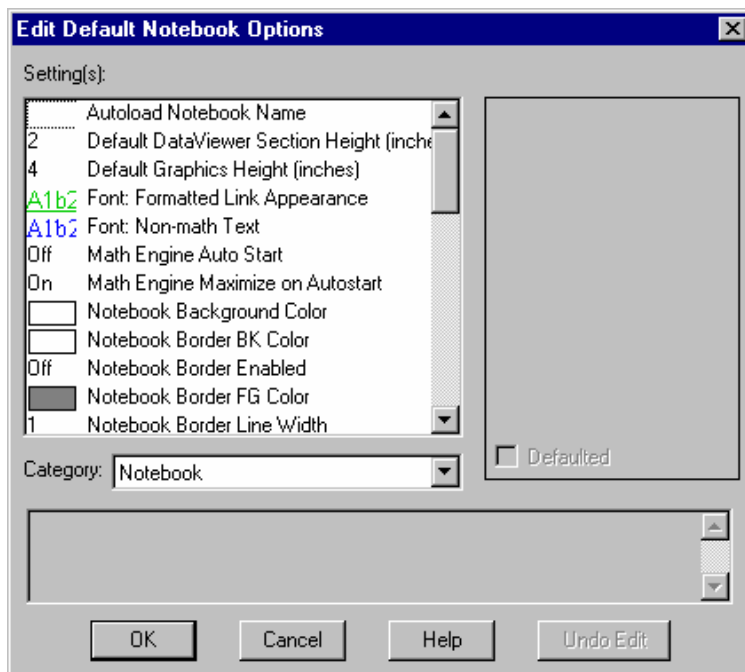
When an error occurs, or you press Interrupt, you may wind up in a Macsyma Break. You may see something like:



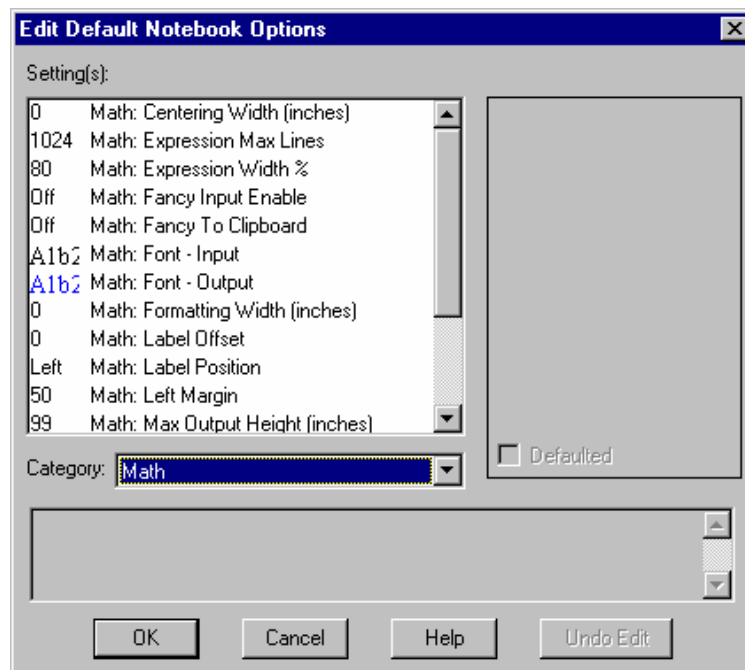
Type :1 followed by Enter to continue. Type :2 followed by Enter to return to a Macsyma c-line.

8.6. How Do I Change Fonts?

Select File - Options or File - Options Default. Select Category Notebook to look for Notebook fonts. Select category Math for math fonts. A Notebook dialog box will look something like:



and a Math category dialog looks something like:



Although you cannot change the font attributes for command line labels, you can change the label offsets. If you want a wider area to display labels, set `Math: Label Margin` to, say 75. You can experiment with other settings.

8.7. How Do I Customize Macsyma?

The file `\macsyma2\mfe\mfe.ini` contains information about directories, variables and attribute settings needed by the Macsyma Front End. You can change this information by selecting `File - Options` or `File - Options Default` from the Macsyma Front End. Some attributes may be available only when Expert Mode is on.

You can place Macsyma commands in the file `mac-init.mac` that affect the Macsyma Math Engine.

This file should be located in the user's home directory specified by the Macsyma command `user_homedir_pathname()`. One common command is `showtime:true`. In Macsyma 2.2 for Windows NT, `mac-init.mac` should

be located in the user's home directory which is usually `\users\default`. In Windows 95 or Windows 3.X, the home directory is usually `\macsyma2\user`.

8.8. How Can I Make a PostScript File From a Graphic?

You can make a PostScript file from a graphic by printing the graphic to a file using a PostScript printer driver. Install a standard Windows PostScript printer driver and configure it to print to a file. You can do this even though no PostScript printer is physically attached to your PC or network. Windows will prompt you for a file name when you print.

You may need to configure the printer driver to print a PS file or an EPS file.

8.9. Is My Calculation Taking Too Long?

Sometimes, a calculation takes an unexpectedly long time. Delays may occur when you calculate large intermediate symbolic or numeric expressions, such as trying to invert a symbolic matrix. It is often useful to estimate the size of expressions that you expect to get in advance. For a symbolic matrix inversion, it is usually not practical to invert a matrix larger than about 7×7 . Inverting numerical matrices is limited only by the resources on your machine.

Macsyma has a status line at the bottom of the screen. A flashing * on the status line indicates that garbage collection is in progress. If you see lots of rapid flashing, but no screen output, the math engine is still calculating.

Math calculations in Macsyma 2.2 may be interrupted, but the display of sections in the Macsyma Front End cannot be interrupted. Macsyma will not allow keyboard or mouse gestures to be processed until the display has finished. Large sections often take a while to display. Don't assume there is a problem with Macsyma or Windows if you must wait for a response. If you suspect a problem, please try to repeat the calculation by terminating the command line with a \$ to suppress the display. Or you can set the Macsyma option variable `SHOWTIME:TRUE` to indicate a temporal boundary between the math computation and the display.

8.10. How Can I Print a Section From a Notebook?

Macsyma 2.2 prints can print selected sections or all sections of a notebook selected by pages. Large or complex graphics, especially 3D graphics, may not print correctly if you do not have enough printer memory.

8.11. How Can I Get More On-Line Help Using Macsyma?

Macsyma has over 1000 on-line demonstrations and examples. You can find them by looking in MathHelp, MathTips, or Help - Search. Click on Help - Macsyma Demos. Or you can use the MathHelp browser to look for topics. Many, but not all, Macsyma commands have examples. Here is a short list of Macsyma Demos and topic areas.

Topic Area	Description
DEMOS_ARITHMETIC	Demonstrations of different types of numbers
DEMOS_ALGEBRA	Mathematics based on finite processes only
DEMOS_LINEAR_ALGEBRA	Linear algebra and matrices
DEMOS_CALCULUS	Limits, differentiation, integration, integral transforms, vector and tensor calculus
DEMOS_DIFF_EQUATIONS	Ordinary and partial differential equations
DEMOS_ECONOMICS	Cash flows, derivative securities
DEMOS_GEOMETRY	Mathematics with parallelism, lengths, angles
DEMOS_NUMERICAL_ANAL	Roots of equations, integration, statistics, least squares, special functions
DEMOS_PHYSICS	Mechanics, electromagnetism, elasticity, waves, general relativity, quantum mechanics
DEMOS_PLOTTING	2D and 3D graphics, animation
DEMOS_UTILITIES	Rule-based pattern matching, facts database, debugging and metering, translation to other languages, and other programming features.

9. FURTHER ASSISTANCE

There are several examples of notebooks in the `macsyma2\demo` directory. In particular, look at the notebook in `orthcor3.mfe`. It contains interactive examples of graphics that you can re-execute.

Macsyma's on-line documentation is very rich and can often pinpoint the information you need. Macsyma's printed documentation provides similar information in a hard copy format

- The *Macsyma User's Guide* provides a tutorial introduction to Macsyma's most widely used capabilities.
- The *Macsyma Mathematics and System Reference Manual* provides a comprehensive summary of Macsyma's mathematical capabilities. It also describes the non-mathematical aspects of Macsyma, including programming facilities of the Macsyma environment.
- The *Scientific Notebook Interface Reference Manual* comprehensively describes non-mathematical aspects of Macsyma, including the user interface, and the programming facilities of the Macsyma environment.
- The *Scientific Graphics Reference Manual* comprehensively describes non-mathematical aspects of Macsyma graphics commands and data exchange facilities in the `mfe_data` package as well as graphical programming facilities of the Macsyma environment.
- The on-line documentation and release notes should provide information. Be sure to check the Macsyma web page <http://www.macsyma.com> for the very latest information, patches, etc.